

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації
та робототехники
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Розроблення програми-тренажеру для моделювання процесу дистанційного
керування рухом мобільного робота

(тема)

Виконав:
здобувач 2 року навчання,
групи КТРСм-23-2

Ахмад Даніель Халєд

Спеціальності 174 Автоматизація, комп'ютерно-
інтегровані технології та робототехніка

Тип програми Освітньо-професійна

Освітня програма Комп'ютеризовані та
робототехнічні системи

Керівник к.т.н. Сичова О.В.

Допускається до захисту
Зав. кафедри КІТАР

(підпис)

Невлюдов І.Ш.
(прізвище, ініціали)

2025р.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Автоматики і комп'ютеризованих технологій
Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та
робототехніки
Рівень вищої освіти другий (магістерський)
Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
Тип програми Освітньо-професійна
Освітня програма Комп'ютеризовані та робототехнічні системи
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР _____
(підпис)

« 25 » листопада 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Ахмад Даніель Халєд
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення програми-тренажеру для моделювання процесу
дистанційного керування рухом мобільного робота

Затверджена наказом по університету від 25 листопад 2024р. № 1239 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 16.01.2025 р.

3. Вихідні дані до роботи _____

3.1 Спосіб управління програмним тренажером – за допомогою пульта
дистанційного керування

3.2 Інтерфейс передачі даних – бездротовий, радіочастотний

3.3 Реалізувати функцію відтворення властивостей руху мобільного робота
на карті місцевості

4. Перелік питань, що потрібно опрацювати в роботі

4.1 Аналіз предметної області

4.2 Розробка архітектури автоматизованої системи

4.3 Експериментальні дослідження

4.4 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)

Презентаційний матеріал представлений у форматі презентації PowerPoint (*.ppt) – 12 с. формату А4.

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз літератури за темою кваліфікаційної роботи	10.10.24 – 17.10.24	Виконано
2	Аналіз предметної області	18.10.24 – 30.10.24	Виконано
3	Розроблення архітектури автоматизованої системи	01.11.24 – 16.11.24	Виконано
4	Вибір компонентів для побудови макету	17.11.24 – 10.12.24	Виконано
5	Розробка програмного забезпечення для контролера	11.12.24 – 20.12.24	Виконано
6	Оформлення пояснювальної записки	21.12.24 – 13.01.25	Виконано
7	Подання роботи на перевірку Інтернет-сервісом Unichesk	14.01.25 – 15.01.25	
8	Подання роботи на рецензію	16.01.25 – 17.01.25	
9	Подання роботи на підпис зав. кафедри	17.01.25 – 18.01.25	
10	Подання кваліфікаційної роботи в ЕК	18.01.25	

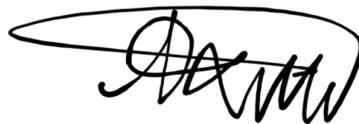
Дата видачі завдання 02.09.2024 р.

Здобувач _____ Ахмад Даніель Халед
(підпис)

Керівник роботи _____ к.т.н. Сичова О.В.
(підпис) (посада, прізвище, ініціали)

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«10» січня 2025 р.

A handwritten signature in black ink, appearing to read 'Ahmad Daniel Khalid', enclosed within a large, horizontal, hand-drawn oval.

Ахмад Даніель Халед

РЕФЕРАТ

Пояснювальна записка: 74 с., 25 рис., 3 дод., 13 джерел.

ПРОГРАМНИЙ ТРЕНАЖЕР, ЦИФРОВИЙ ДВІЙНИК, МОБІЛЬНИЙ РОБОТИЗОВАНИЙ ПРИСТРІЙ, ДИСТАНЦІЙНЕ КЕРУВАННЯ.

Об'єктом дослідження в даній роботі є процес дистанційного керування рухом наземних безпілотних роботизованих пристроїв.

Предмет дослідження – методи моделювання та програмної симуляції руху безпілотного роботизованого транспортного засобу з використанням цифрової моделі заданого об'єкта.

Метою цієї роботи є дослідження підвищення ефективності методу дистанційного керування роботизованим засобом при поєднанні в єдиному програмному інструменті технології моделювання стану мобільного робота за допомогою програмного тренажера та цифрового двійника роботизованого засобу.

В роботі проведено порівняльний аналіз сфер застосування програмних тренажерів та цифрових двійників. Праведно аналіз методів та принципів моделювання поведінки мобільних роботизованих пристроїв з використанням цифрових двійників.

Розроблено програму-тренажер для моделювання поведінки гусеничного мобільного робота. Пульт дистанційного керування виконує роль фізичного інтерфейсу, тоді як програмний тренажер дозволяє безпечно та ефективно тренувати операторів у віддаленому керуванні мобільними роботами.

Також, отримані результати роботи можна віднести до Цілі сталого розвитку 9 «Промисловість, інновації та інфраструктура», а саме п. 9.2 «Забезпечити розширення використання електротранспорту та відповідної мережі інфраструктури».

ABSTRACT

Explanatory note: 74 p., 25 fig., 3 app.,13 sources.

SOFTWARE SIMULATOR, DIGITAL DOUBLE, MOBILE ROBOTIC DEVICE, REMOTE CONTROL.

The object of research in this work is the process of remote control of the movement of ground-based unmanned robotic devices.

The subject of research is methods of modeling and software simulation of the movement of an unmanned robotic vehicle using a digital model of a given object.

The purpose of this work is to study the effectiveness of the method of remote control of a robotic vehicle when combining in a single software tool the technology of modeling the state of a mobile robot using a software simulator and a digital double of a robotic vehicle.

The work provides a comparative analysis of the areas of application of software simulators and digital twins. The analysis of methods and principles of modeling the behavior of mobile robotic devices using digital twins is fair.

A simulator program has been developed to simulate the behavior of a tracked mobile robot. The remote control serves as a physical interface, while the software simulator allows operators to safely and effectively train in remote control of mobile robots.

Also, the results of the work can be attributed to Sustainable Development Goal 9 "Industry, Innovation and Infrastructure", namely item 9.2 "Ensure the expansion of the use of electric transport and the corresponding infrastructure network".

ЗМІСТ

Перелік скорочень	9
Вступ.....	10
1 Аналіз предметної області	12
1.1 Порівняльний аналіз сфер застосування програмних тренажерів та цифрових двійників.....	12
1.2 Аналіз методів та принципів моделювання поведінки мобільних роботизованих пристроїв з використанням цифрових двійників	18
1.3 Висновки по першому розділу.....	26
2 Розробка математичної моделі мобільного робота	28
2.1 Загальні відомості про об'єкт управління.....	28
2.2 Математична модель мобільної гусеничної моделі	30
2.3 Висновки по другому розділу.....	38
3 Архітектура автоматизованої системи.....	39
3.1 Опис архітектури та принципу дії автоматизованої системи	39
3.2 Структурна схема системи автоматичного управління.....	43
3.3 Висновки по третьому розділу	48
4 Експериментальні дослідження.....	49
4.1 Вибір компонентів для побудови макету	49
4.2 Розробка програмного забезпечення для контролера	56
4.3 Розробка програмного забезпечення для персонального комп'ютера ..	59
4.4 Приклад роботи програми	69
4.5 Розрахунок кількості приливу повітря з урахуванням чисельності працюючих	72

4.6 Висновки по четвертому розділу.....	73
Висновки.....	74
Перелік джерел посилань.....	76
Додаток А Апробація результатів наукових досліджень.....	78
Додаток Б Вихідний код програми.....	84
Додаток В Демонстраційний матеріал.....	93

ПЕРЕЛІК СКОРОЧЕНЬ

АП – автоматизований пристрій;

ОУ – об'єкт управління;

ПЗ – програмне забезпечення;

ПД – пропорційне диференційне регулювання;

ШІМ – широтно-імпульсна модуляція;

API – application programming interfaces;

DT – digital twin;

DTA – digital twin aggregate;

DTI – digital twin instance;

DTP – digital twin prototypes;

EKF – extended Kalman filter;

PLM – product lifecycle management;

ROS – robotic operation system.

ВСТУП

У сучасному світі мобільні роботи стають невід'ємною частиною автоматизованих систем, які широко застосовуються в різних галузях – від промислових підприємств до військової сфери та сільського господарства. Однак для ефективного управління мобільними роботами, особливо в складних та динамічних середовищах, необхідна підготовка операторів та інженерів, що включає не лише теоретичні знання, але й практичні навички. У цьому контексті програмні тренажери для моделювання процесу дистанційного керування мобільними роботами стають актуальним та важливим інструментом.

Програмні тренажери для дистанційного керування мобільними роботами дають можливість моделювати реальні ситуації без необхідності використання фізичних пристроїв.

Програмні тренажери та цифрові двійники мають спільну мету – підвищення ефективності роботи та оптимізація процесів керування. Однак вони застосовуються в різних контекстах: тренажери використовуються для навчання й тестування в симульованих умовах, тоді як цифрові двійники інтегруються з реальними системами для управління, прогнозування та аналізу. Обидві технології мають величезний потенціал для розвитку робототехніки, промислової автоматизації та інших галузей, де важлива точність, безпека та ефективність процесів.

Мета кваліфікаційної роботи – дослідження підвищення ефективності методу дистанційного керування роботизованим засобом при поєднанні в єдиному програмному інструменті технології моделювання стану мобільного робота за допомогою програмного тренажера та цифрового двійника роботизованого засобу.

Об'єктом дослідження в даній роботі є процес дистанційного керування рухом наземних безпілотних роботизованих пристроїв.

Предмет дослідження – методи моделювання та програмної симуляції руху безпілотного роботизованого транспортного засобу з використанням цифрової моделі заданого об'єкта.

Для досягнення мети необхідно вирішити наступні завдання:

- виконати порівняльний аналіз сфер застосування програмних тренажерів та цифрових двійників;

- провести аналіз методів та принципів моделювання поведінки мобільних роботизованих пристроїв з використанням цифрових двійників;

- розробити архітектуру автоматизованої системи дистанційного керування рухом мобільного роботизованого пристрою з використанням програмного тренажера та пульта дистанційного керування;

- виконати побудову алгоритму та розробити програму для проведення експериментальних досліджень;

- виконати експериментальні дослідження для підтвердження правильності теоретичних рішень.

Пояснювальну записку оформлено згідно з вимогами ДСТУ 3008:2015 [1] та рекомендаціями [2].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Порівняльний аналіз сфер застосування програмних тренажерів та цифрових двійників

Зазвичай розробникам у сфері робототехніки потрібно протестувати, перевірити та перевірити динаміку робота в певних середовищах [7]. Сьогодні доступність віртуальних інструментів дозволяє розробникам створювати віртуальні робочі середовища та проводити цілеспрямоване моделювання динаміки роботів. Цей підхід значно скорочує час, необхідний для підготовки конкретного контенту для робота (дизайн, розгортання, середовище), а потім і для налагодження динаміки робота.

Програмний тренажер – це віртуальна симуляція, яка використовується для навчання, тестування та вдосконалення навичок роботи з різними технічними системами, такими як роботи, обладнання або транспортні засоби [7]. Тренажери дозволяють імітувати роботу систем у контрольованому середовищі, що знижує ризики помилок та втрат, забезпечуючи безпечне навчання операторів.

На рисунку 1.1 показано приклад інтерфейсу програмного тренажеру [8].

Використовуючи програмне забезпечення для моделювання для програмування робота, важливо пам'ятати, що жодне моделювання не зможе точно відобразити те, що насправді станеться, коли ви запуснете програму зі справжнім роботом. Рух кабелів і шлангів не моделюється більшістю програмних середовищ, тому можливо, що запрограмований рух може зачепити або розтягнути кабелі, спричинивши збій.

Одним із факторів, який сприяє кращому моделюванню, є перехід від симуляції на основі часу до моделювання на основі подій. Раніше розробникам симуляторів доводилося вгадувати, скільки часу займе певна подія, але

насправді ці робототехнічні системи працюють не так. Вони працюють у режимі, заснованому на подіях, і якщо ми зможемо змоделювати систему так, як вона насправді працює на підлозі, забезпечивши запуск і зупинку цих подій і координацію одна з одною, ми також досягнемо вищого рівня точності між моделюванням і реальне виконання цієї програмної моделі в цеху [8].

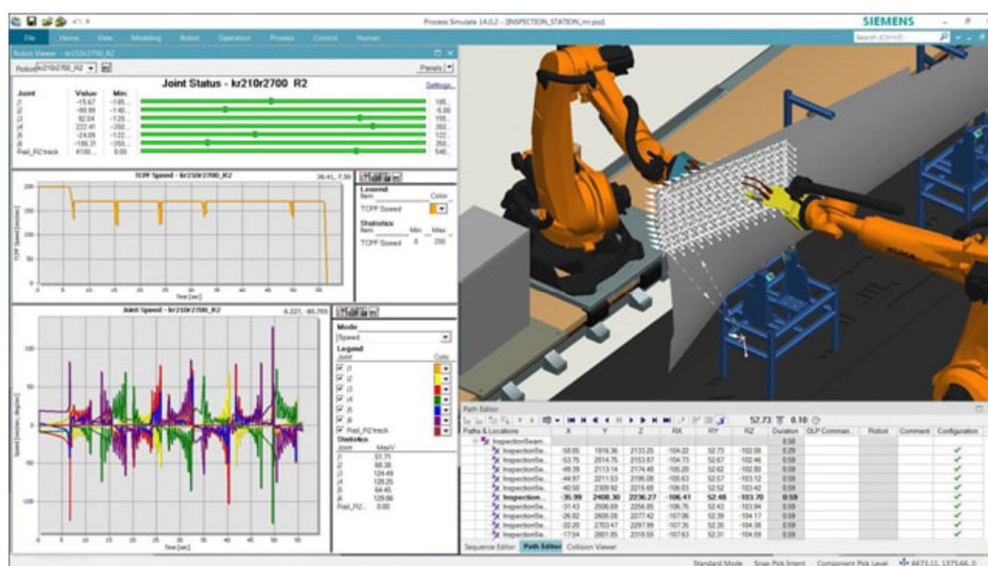


Рисунок 1.1 – Приклад інтерфейсу програмного тренажеру

Програмні тренажери для дистанційного керування мобільними роботами дають можливість моделювати реальні ситуації без необхідності використання фізичних пристроїв.

Оператори можуть практикуватися в керуванні мобільними роботами в умовах, де ризики аварій, ушкоджень техніки або операторських помилок зведені до мінімуму. Використання програмного забезпечення дозволяє уникнути високих витрат на придбання та обслуговування реальних роботів під час навчання. Це значно знижує фінансове навантаження на компанії та навчальні заклади.

За допомогою тренажерів можна імітувати найрізноманітніші умови, з якими оператори можуть зіткнутися в реальному світі. Це може бути як

складний рельєф, так і динамічні умови середовища, наприклад, змінна швидкість вітру, перешкоди на шляху тощо. Програмні тренажери дозволяють миттєво аналізувати дії користувача, що дає можливість операторам швидко навчатися на своїх помилках і покращувати свої навички.

Окрім навчання операторів, програмні тренажери є важливими для розробників і інженерів, які працюють над створенням та вдосконаленням алгоритмів дистанційного керування роботами. Моделювання на тренажері дає можливість тестувати алгоритми в симульованому середовищі до їхнього впровадження на реальні пристрої. Це дозволяє значно знизити ймовірність помилок та підвищити ефективність роботи системи управління.

Зокрема, програмні тренажери дозволяють проводити тестування алгоритмів у середовищах, які важко або неможливо змоделювати фізично, наприклад, при роботі у вакуумі або на різних планетарних поверхнях у космічних місіях. Також тренажери можуть моделювати несприятливі погодні умови або екстремальні температури, що є критичним для військових або рятувальних операцій.

Програмні тренажери стають невід'ємною частиною навчання у закладах вищої освіти, де готують фахівців з робототехніки, автоматизації та інженерії. Вони дозволяють студентам отримувати практичні навички керування мобільними роботами без ризику ушкодження обладнання та з мінімальними витратами. Такі тренажери дають можливість вивчати різні аспекти робототехніки – від програмування рухів до моделювання поведінки у змінних середовищах.

Використання програмних тренажерів дає змогу майбутнім фахівцям вивчати нові технології в інтерактивному середовищі, що значно покращує їхню підготовку. Завдяки цим тренажерам студенти можуть виконувати індивідуальні та групові завдання, аналізувати власні помилки і вдосконалювати навички перед роботою з реальними пристроями.

Цифровий двійник (Digital Twin) – це цифрова модель фізичного об'єкта або системи, яка відображає її стан у реальному часі, збираючи дані з

фізичного прототипу та відтворюючи їх у віртуальному середовищі [9]. Цифровий двійник дозволяє не тільки спостерігати за роботою системи, але й прогнозувати її поведінку, тестувати сценарії та оптимізувати процеси.

Приклад інтерфейсу цифрового двійника виробничої ділянки наведено на рисунку 1.2.



Рисунок 1.2 – Приклад інтерфейсу цифрового двійника виробничої ділянки

В роботі [7] проводиться дослідження використання графічного моделювання в поєднанні з програмуванням, що значно підвищило дизайн, контроль і продуктивність роботизованих програм, особливо у виробництві. Під час проектування та керування робототехнічними системами можна використовувати комп'ютерну графіку, щоб оцінити, як робот реагуватиме на виробниче середовище та працюватиме у ньому до того, як буде понесено значний час, капітал і витрати. В даній роботі представлено програму керування на основі ROS для роботизованої платформи з використанням симулятора Gazebo 3D, який дозволить користувачеві бачити та розуміти

симуляцію та переміщення мобільного робота через попередньо визначену карту перед використанням у реальному світі.

Автори роботи [10] визнають цифровий двійник (DT) ключовим інструментом для автоматизації лабораторій і розумних фабрик, який може візуалізувати, симулювати та контролювати свій аналог у міру розвитку цифровізації. Щоб забезпечити перевірку системи, потрібна точна синхронізація стану між DT і фізичним близнюком (PT). Точніше, позиція в реальному часі розглядається як життєво важливий показник при впровадженні мобільних роботів.

У статті [10] пропонуємо систему DT для випробувального стенда мобільного робота за допомогою централізованої інтегрованої системи локалізації. Інтегрована система локалізації реалізована за допомогою бортового датчика робота, зображень з камери монітора та розширеного фільтра Калмана (ЕКФ), який дозволяє оцінювати стан PT у реальному часі та передавати його на контролер DT. Крім того, система DT інтегрує функції 3D-симуляції та забезпечує порт для фізичного керування роботом. Запропонований підхід пропонує рішення для синхронізації стану, яке не покладається на складні датчики або здатність локалізації окремого робота, що вказує на потенціал інтеграції роботів з різними технологічними можливостями.

Обидва інструменти – програмний тренажер і цифровий двійник – використовуються для віртуального моделювання, але їхні цілі та функції суттєво різняться.

Програмні тренажери орієнтовані, насамперед, на навчання та розвиток навичок. Вони дозволяють імітувати реальні умови роботи та навчати операторів або інженерів без необхідності використання фізичних пристроїв. Наприклад, оператори можуть навчитися керувати мобільними роботами або іншими складними системами без ризику ушкоджень техніки.

Цифрові двійники, навпаки, спрямовані на забезпечення точного відображення стану реальної системи в режимі реального часу. Це не просто

симуляція, а інтегрована частина керування фізичним об'єктом. Цифрові двійники дозволяють відстежувати поведінку обладнання або процесу, прогнозувати потенційні проблеми, проводити оптимізацію і навіть прогнозувати майбутні сценарії його використання на основі реальних даних.

Таким чином, якщо програмні тренажери використовуються переважно для навчальних цілей, то цифрові двійники знаходять своє застосування в управлінні та оптимізації реальних систем на основі аналітики та прогнозування.

Програмні тренажери сприяють безпечному навчанню віртуальному керуванню реальними системами. Наприклад, тренажери для мобільних роботів дозволяють оператору тренуватися в управлінні та виконанні складних операцій без ризику пошкодження обладнання, що є особливо важливим у військовій, медичній або промисловій сферах. Тренажери також можуть симулювати екстремальні або рідкісні ситуації, в яких оператори не мали б змоги тренуватися в реальному житті.

Цифрові двійники, своєю чергою, сприяють підвищенню ефективності операцій у реальному часі. Наприклад, двійник виробничого обладнання може постійно моніторити стан механізмів і попереджати про можливі несправності ще до того, як вони виникнуть. Це дозволяє вживати запобіжних заходів, уникати простоїв і забезпечувати більш ефективне використання ресурсів.

Одна з ключових різниць між програмними тренажерами і цифровими двійниками полягає в тому, як вони використовують дані.

Програмний тренажер забезпечує симуляцію фіксованих сценаріїв на основі заданих параметрів. Це означає, що оператори можуть моделювати різні ситуації, але система працює на основі наперед визначених алгоритмів і не враховує реальні дані з фізичних пристроїв. Проте це дає змогу швидко тестувати нові ідеї, не вдаючись до фізичних змін у реальних системах.

Цифровий двійник використовує реальні дані для моніторингу системи в режимі реального часу [9, 10]. Завдяки постійній інтеграції з фізичним об'єктом, цифровий двійник дозволяє не лише слідкувати за поточним станом

системи, але й прогнозувати її майбутній стан. Це дозволяє виявляти потенційні проблеми або оптимальні шляхи підвищення ефективності, аналізуючи великі обсяги даних.

Програмні тренажери мають обмеження у тому, що вони моделюють тільки ті сценарії, які були заздалегідь визначені. Це знижує їх гнучкість, оскільки будь-які зміни в сценарії потребують додаткового програмування.

Цифрові двійники є більш гнучкими та масштабованими. Оскільки вони інтегруються з реальними даними, вони можуть динамічно змінюватися разом із фізичною системою, підлаштовуючись під нові умови та сценарії. Цифрові двійники можуть охоплювати не лише окремі машини, а й цілі заводи або логістичні мережі, забезпечуючи ефективне управління на всіх рівнях.

Таким чином, програмні тренажери та цифрові двійники мають спільну мету – підвищення ефективності роботи та оптимізація процесів керування. Однак вони застосовуються в різних контекстах: тренажери використовуються для навчання й тестування в симульованих умовах, тоді як цифрові двійники інтегруються з реальними системами для управління, прогнозування та аналізу. Обидві технології мають величезний потенціал для розвитку робототехніки, промислової автоматизації та інших галузей, де важлива точність, безпека та ефективність процесів.

1.2 Аналіз методів та принципів моделювання поведінки мобільних роботизованих пристроїв з використанням цифрових двійників

Без створення цифрових двійників виробів неможливе впровадження сучасної технології PLM (Product Lifecycle Management, управління життєвим циклом виробу) [9]. ІоТ і PLM – невід’ємні атрибути «розумної фабрики» (Smart Factory). Її характерна риса – формування і використання цифрової моделі матеріальних потоків, тобто, цифрового двійника вже не окремого виробу, а виробничої системи.

Усі зазначені технології є підходами до впровадження концепції Четвертої індустріальної революції (Industry 4.0). У традиційній промисловості для досягнення необхідних характеристик виробу проводяться численні натурні випробування. Натомість, Індустрія 4.0 передбачає багаторазові випробування за допомогою цифрового двійника, що дозволяє проходити натурні випробування з першого разу.

Цифровий двійник виробу включає:

- геометричну та структурну модель об'єкта;
- набір розрахункових даних деталей, вузлів і виробів загалом;
- математичні моделі, що описують усі фізичні процеси у виробі;
- інформацію про технологічні процеси виготовлення та збирання окремих елементів і виробу в цілому;
- систему керування життєвим циклом виробу.

Цифровий двійник об'єкта є засобом доступу до інформації про життєвий цикл та єдиним інтерфейсом до неї. Цифрові двійники можуть бути створені для будь-якої сутності, яка цікавить підприємство. Цифровий двійник складається з даних, цифрових моделей та сервісних інтерфейсів (рис. 2.3).

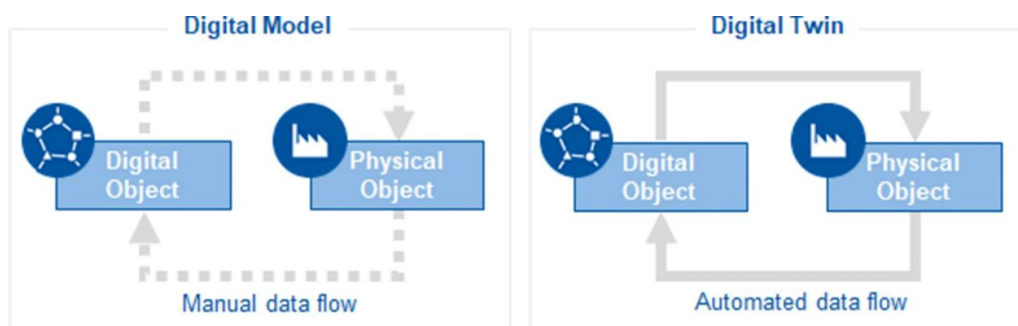


Рисунок 1.3 – Структура цифрового двійника об'єкту

Цифровий двійник використовується на всіх етапах життєвого циклу виробу, починаючи з проектування і закінчуючи експлуатацією та утилізацією. На стадії попереднього проектування створюються різні

комп'ютерні моделі виробу для оцінки та вибору оптимальних технічних рішень.

На етапі технічного проектування обраний варіант моделі уточнюється та допрацьовується з використанням елементних моделей. Модель дозволяє оптимізувати взаємодію компонентів виробу, враховуючи робочі режими та зовнішні умови. Ця модель вже може вважатися цифровим двійником, який поступово доповнюється деталями.

На стадії виробництва цифрова модель допомагає визначити необхідні допуски для досягнення бажаних характеристик виробу та забезпечення його надійності. Модель також дозволяє оперативно виявляти причини несправностей під час тестування.

На етапі експлуатації цифровий двійник може бути адаптований для надання зворотного зв'язку, що допомагає вдосконалювати проектування та виробництво, діагностувати й прогнозувати несправності, підвищувати ефективність роботи і реагувати на запити користувачів.

Цифрові двійники класифікуються за такими типами:

- прототипи цифрових двійників (Digital Twin Prototypes, DTP);
- екземпляри цифрових двійників (Digital Twin Instance, DTI);
- агреговані цифрові двійники (Digital Twin Aggregate, DTA).

DTP містить усі необхідні дані для опису та створення фізичних версій виробу, включаючи геометричну модель, технічні вимоги та інші параметри. DTP можна вважати постійною віртуальною моделлю виробу.

DTI описує конкретний фізичний екземпляр виробу, який підтримує зв'язок із цифровим двійником протягом усього терміну експлуатації. Цей тип двійника базується на DTP і додатково включає дані про виробництво та експлуатацію, такі як історія виготовлення, використані матеріали, статистика несправностей і ремонтів. DTI оновлюється відповідно до змін фізичного виробу.

DTA є інформаційною системою, яка керує фізичними екземплярами виробів, забезпечуючи доступ до всіх їхніх цифрових двійників.

Класифікація цифрових двійників у виробничих системах:

- цифровий двійник усієї виробничої системи;
- цифровий двійник окремої виробничої лінії;
- цифровий двійник конкретного елемента виробничої лінії.

Рівень абстракції цифрового двійника залежить від умов використання, для яких він розробляється (рис. 1.4).

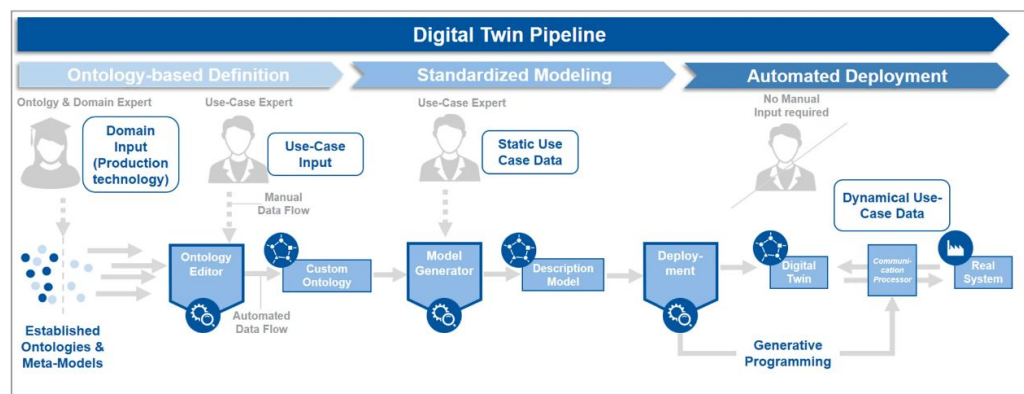


Рисунок 1.4 – Три фази наскрізного цифрового подвійного конвеєра для застосування в кіберфізичних системах

Індивідуальний цифровий двійник є цілісною сутністю, яка не потребує подальшого поділу. Наприклад, коробка передач або двигун роботизованого засобу в умовах моніторингу можуть розглядатися як єдиний об'єкт. Об'єднання окремих цифрових двійників для створення складного цифрового двійника представлено як вертикальне розширення, що демонструє зростання кількості сутностей від одиничної до множинної.

Складений цифровий двійник являє собою сукупність окремих цифрових двійників, що описують об'єкт, який складається з кількох частин або компонентів. Така композиція може відбуватися на різних рівнях. Наприклад, виробнича комірка є складним об'єктом, чий цифровий двійник

складається з цифрових двійників пристроїв, що працюють у межах цієї комірки.

У складеному цифровому двійнику зв'язки між елементами можуть бути наступними:

– ієрархічні зв'язки: як і у фізичних аналогів, набір цифрових двійників компонентів об'єднується в цифровий двійник устаткування; кілька таких двійників формують цифровий двійник виробничої лінії, а набір таких ліній може стати цифровим двійником заводу;

– асоціативні зв'язки: як і у реальних об'єктів, між цифровими двійниками можуть існувати асоціації. Наприклад, цифровий двійник конвеєру може бути пов'язаний з двійниками процесів складання на цифровому заводі;

– однорангові зв'язки (peer-to-peer): такі зв'язки спостерігаються між рівноправними елементами в групах обладнання схожого типу, що виконують аналогічні функції. Наприклад, кроковий двигун, де двигуни формують єдиний цифровий двійник рухомої платформи.

Семантика похідної онтології зображена на рис. 1.5 [11].

На приклад, замовлення складається з різних продуктів, які мають бути виготовлені. Усі ресурси можна розрізнити за типами та екземплярами. Наприклад, продукт відповідає типу продукту, що означає, що застосовуються його підмоделі.

Крім того, описані типи етапів процесу та типи деталей застосовуються, створюються та призначаються продукту. Відповідно до [11], ресурси можна розділити на географічні ресурси, які в цьому випадку представлені онтологією VOT, людські ресурси та машинні ресурси, які можуть бути мобільними або стаціонарними. Процес узгодження системою керування може бути здійснено шляхом зіставлення певного кроку процесу з ресурсом, на якому він буде виконуватися.

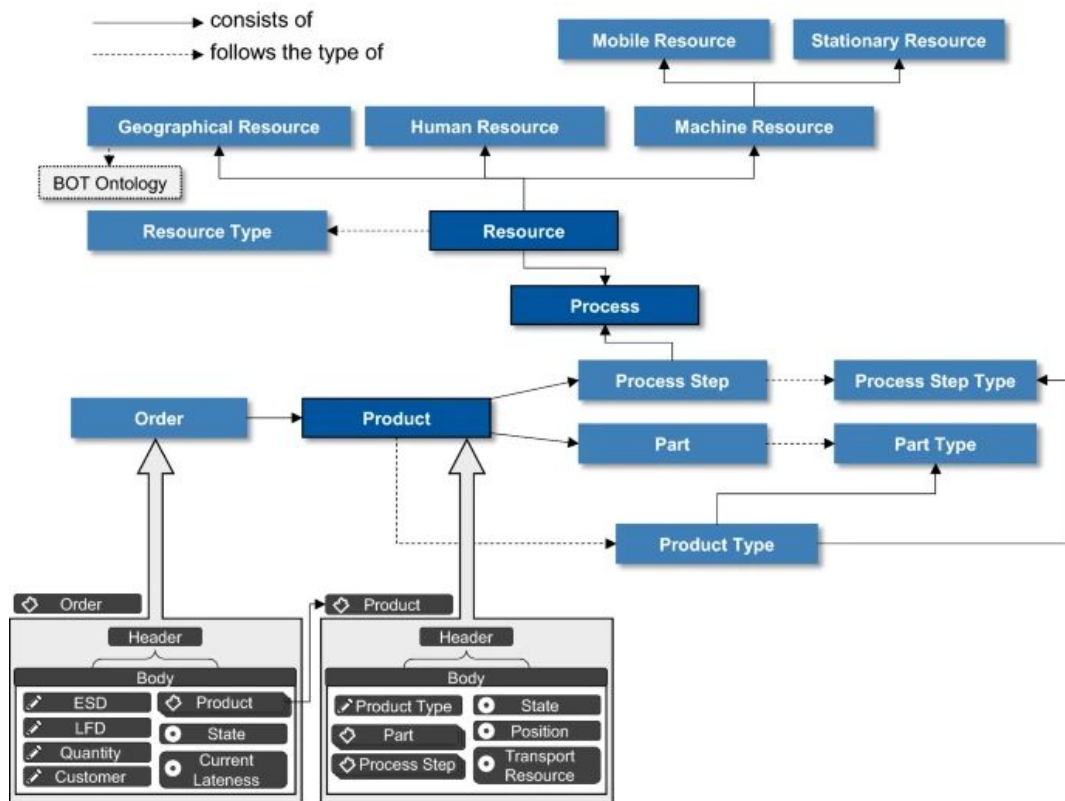


Рисунок 1.5 – Приклад предметно-орієнтованої онтології, що включає встановлені онтології та метамоделі при створенні цифрового двійника

Комбінація ресурсів (окрім машин також можуть знадобитися оператори та інструменти) і крок процесу призводить до процесу. Усі відображені об'єкти є компонентами, які самі складаються з підмоделей, таких як компоненти, змінні, параметри та функції. Це показано як приклад для замовлення та продукту, де замовлення містить продукти, змінні, як-от поточна затримка, і параметри, як-от відповідний клієнт і найраніша дата початку (ESD). Сам включений компонентний продукт має заголовок із представленням характеристик і тіло з підмоделями, як-от включені частини та етапи процесу, як компоненти, і поточну позицію як змінну.

Зазвичай інформація про об'єкт розподілена по різних інформаційних ресурсах, які створюються та підтримуються різними організаціями. Це може порушувати цілісність інформаційного потоку протягом усього життєвого

циклу об'єкта, оскільки такі ресурси не завжди коректно взаємодіють між собою. Частина інформації може дублюватися або містити суперечності, а інша – бути відсутньою. Це призводить до затримок через необхідність пошуку даних, їх впорядкування та встановлення логічних зв'язків, що вимагає значних витрат часу. Виникає також ризик конфліктів в оперативній інформації, що може призвести до неефективних рішень. Крім того, окремі інформаційні сховища ускладнюють впровадження сучасних технологій, таких як розширена аналітика та штучний інтелект, які потребують доступу до великих масивів даних.

Цифрові двійники дозволяють подолати проблему розрізнених інформаційних сховищ (рис. 1.6).



Рисунок 1.6 – Розгортання агента керування з майстром зв'язку

Вони виступають як посередник, який централізує збір даних про кожен об'єкт та через інтеграційні інтерфейси, такі як API (Application Programming Interfaces), забезпечує доступ до цієї інформації для різних бізнес-процесів. Це сприяє прийняттю більш обґрунтованих рішень, забезпечує загальне розуміння поточного стану об'єктів та знижує загальні витрати на життєвий цикл експлуатації та обслуговування виробничих об'єктів.

Щоб представляти реальні об'єкти в динамічному вигляді, цифрові двійники повинні бути пов'язані з фізичними оригіналами, часто в режимі реального часу, для збору та впорядкування даних відповідних об'єктів (рис. 1.7). У цифрових двійниках мають бути присутні аналітичні та розрахункові моделі, які дозволяють описувати, діагностувати, передбачати та моделювати стани і поведінку фізичних систем.

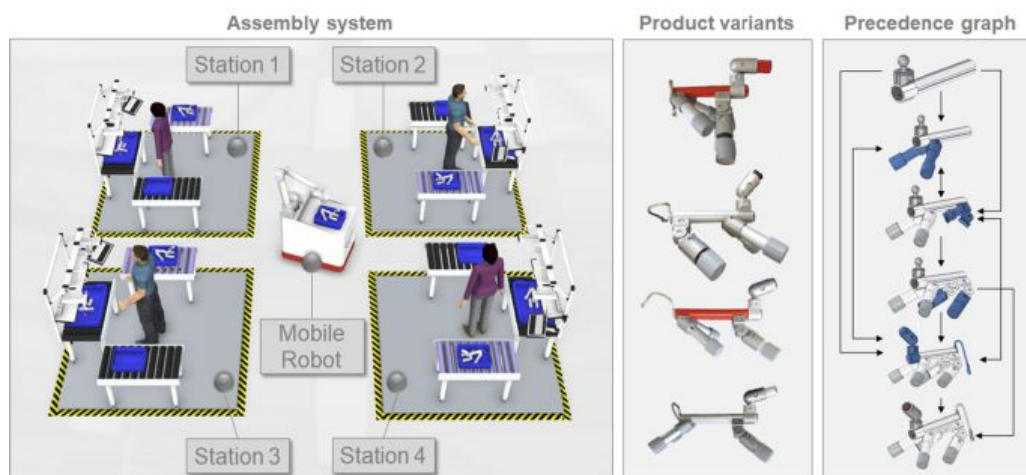


Рисунок 1.7 – Цифровий двійник реального процесу із застосуванням складальних операцій

Отримані внаслідок аналізу висновки можуть бути інтегровані з бізнес-процесами, що сприятиме оптимізації виробничих операцій. Для цього необхідно, щоб розробка цифрових двійників передбачала наявність сервісних інтерфейсів, через які інтелектуальні промислові застосунки могли б отримувати доступ до даних та аналітичних результатів.

Цифровий двійник складається з даних, розрахункових моделей і сервісних інтерфейсів (рис. 1.7), подібно до об'єктно-орієнтованого програмування, де є дані, методи та інтерфейси. Для коректного відображення стану об'єкта, цифровий двійник має містити інформацію про свою фізичну копію. Дані можуть охоплювати весь життєвий цикл об'єкта, включаючи

проектування (специфікації, технічні характеристики), виробництво (дані про працівників, матеріали, контроль якості), експлуатацію (налаштування, архівні дані) та навіть утилізацію.

Для точного прогнозування стану та поведінки реальних об'єктів цифрові двійники використовують розрахункові та аналітичні моделі. Вони можуть базуватися на фізичних або хімічних процесах, інженерних методах, імітації, статистиці, машинному навчанні та штучному інтелекті. Просторові моделі та моделі доповненої реальності також можуть сприяти розумінню поточних робочих станів.

Для взаємодії з іншими промисловими застосунками чи цифровими двійниками потрібні сервісні інтерфейси, які забезпечують доступ до даних. Незважаючи на різноманітність форм і змісту об'єктів, кожен цифровий двійник має містити базові конструкції з загальними атрибутами даних і моделей, що забезпечує уніфіковане звернення до них.

1.3 Висновки по першому розділу

В результаті виконання першого розділу кваліфікаційної роботи проведено порівняльний аналіз сфер застосування програмних тренажерів та цифрових двійників. Тренажери дозволяють імітувати роботу систем у контрольованому середовищі, що знижує ризики помилок та втрат, забезпечуючи безпечно навчання операторів.

Використання програмних тренажерів дає змогу майбутнім фахівцям вивчати нові технології в інтерактивному середовищі, що значно покращує їхню підготовку. Завдяки цим тренажерам студенти можуть виконувати індивідуальні та групові завдання, аналізувати власні помилки і вдосконалювати навички перед роботою з реальними пристроями.

Проведено аналіз методів та принципів моделювання поведінки мобільних роботизованих пристроїв з використанням цифрових двійників.

наведено структуру цифрового двійника об'єкту. Надано приклад предметно-орієнтованої онтології, що включає встановлені онтології та метамоделі при створенні цифрового двійника.

2 РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ МОБІЛЬНОГО РОБОТА

2.1 Загальні відомості про об'єкт управління

В якості об'єкта управління розглянемо гусеничну мобільну роботизовану платформу. Мобільний гусеничний робот є механізмом, який пересувається за допомогою гусеничної ходової частини, забезпечуючи високу прохідність на нерівних і складних поверхнях (рис. 2.1). Гусеничні роботи широко застосовуються в умовах, де необхідно забезпечити стабільний рух на пересіченій місцевості, піску, снігу або навіть болотах [12].



Рисунок 2.1 – Приклад гусеничного робота-дослідника

З точки зору математичної моделі, гусеничний робот має низку особливостей, які необхідно враховувати для його успішного управління та контролю.

Основою математичної моделі є кінематичні та динамічні рівняння, які описують рух робота. Кінематична модель визначає залежність між

швидкостями руху робота і його положенням у просторі. Зокрема, робот може переміщуватися в площині за допомогою двох гусениць, які рухаються з різними швидкостями. Для опису положення робота використовують його координати x та y у площині, а також кут орієнтації θ , який показує напрям руху відносно початкової осі.

Кінематика гусеничного робота базується на різниці швидкостей лівої та правої гусениці. Якщо обидві гусениці рухаються з однаковою швидкістю, робот рухається прямо. Якщо швидкості різні, робот починає обертатися навколо своєї осі. Лінійна швидкість v і кутова швидкість ω можуть бути визначені через швидкості гусениць v_L і v_R , використовуючи такі рівняння:

$$v = \frac{v_L + v_R}{2}, \quad (2.1)$$

$$\omega = \frac{v_L - v_R}{b}, \quad (2.2)$$

де b – ширина бази робота (відстань між гусеницями).

Динамічна модель робота враховує сили та моменти, що діють на нього під час руху. Це дозволяє описати, як сили тяги, тертя, та моменти інерції впливають на рух робота. Лінійне прискорення робота визначається сумою сил тяги від кожної гусениці, а обертальний момент – різницею цих сил.

Для гусеничного робота важливим є контакт з поверхнею. Гусениці забезпечують велику площу контакту з поверхнею, що збільшує силу тертя і покращує стійкість робота. Модель тертя може бути наближено описана за допомогою закону Кулона:

$$F_{\text{тертя}} = \mu N, \quad (2.3)$$

де μ – коефіцієнт тертя;

N – нормальна сила, що діє на гусениці від ваги робота.

Управління гусеничним роботом реалізується шляхом диференціального приводу, де окремо регулюються швидкості кожної з гусениць. Це дає можливість виконувати як прямолінійний рух, так і повороти з різним радіусом. Наприклад, для повороту на місці одна гусениця повинна рухатися вперед, а інша назад із тією ж швидкістю, що створює обертання без переміщення в просторі.

Однією з ключових задач в управлінні гусеничним роботом є стабілізація його руху на складних поверхнях. Через великі сили тертя і нерівномірність контакту з поверхнею можуть виникати додаткові перешкоди для керованого руху. Система управління повинна враховувати зміни поверхні і адаптувати швидкості гусениць для досягнення бажаного напрямку і швидкості руху.

Окрім того, у складних умовах, таких як слизькі поверхні, може бути реалізовано контроль тяги та корекція курсу для уникнення прослизання гусениць. Це вимагає використання сенсорів для моніторингу швидкостей руху, сили тяги і кутової орієнтації робота.

2.2 Математична модель мобільної гусеничної моделі

Для побудови математичної моделі рух гусеничної мобільної платформи представляється у вигляді просторового руху твердого тіла, миттєвий стан якого оцінюється векторами швидкості поступального руху центру мас V [12]

$$V = V_x i + V_y j + V_z k, \quad (2.4)$$

та кутової швидкості обертання щодо центру мас Ω

$$\Omega = \omega_x i + \omega_y j + \omega_z k. \quad (2.5)$$

У цій моделі використовуються три різні системи координат (рис. 2.2):

– нерухома система координат $O_2X_2Y_2Z_2$ призначена для опису заданих дорожніх ґрунтових умов руху, причому її початок збігається з базовою точкою рельєфу;

– рухома система координат $O_1X_1Y_1Z_1$ переміщується з центром мас, причому її початок збігається з центром мас, а осі O_1X_1 , O_1Y_1 , O_1Z_1 паралельні відповідним осям системи координат $O_2X_2Y_2Z_2$;

– пов'язана система координат $OXYZ$ використовується для математичного опису динаміки руху, причому її центр завжди збігається з центром мас, а осі OX , OY , OZ збігаються з головними центральними осями інерції мобільного робота.

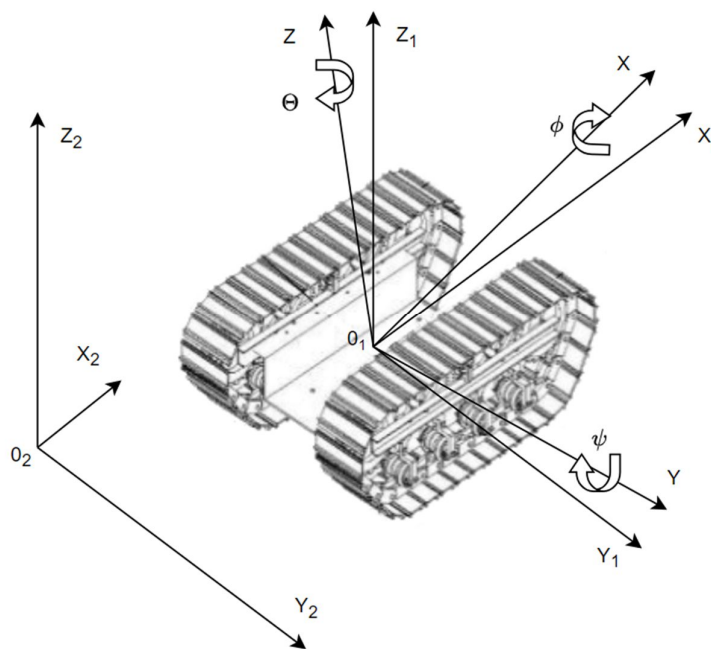


Рисунок 2.2 – Системи координат для опису руху гусеничного мобільного робота

Форма рівнянь динаміки руху мобільного робота, записаних у рухомій системі координат, досить зручна для подальшого математичного моделювання при досить повному відображенні процесів взаємодії тіла, що рухається, і зовнішнього середовища.

Положення мобільного робота у просторі у будь-який момент часу визначається взаємним розташуванням рухомої та пов'язаної систем координат, які можна описати за допомогою кутів Ейлера:

- θ – кут рискання;
- ψ – кут диферента;
- φ – кут крену.

Загалом, розглянута модель динаміки мобільного робота є алгоритмічно досить складною, дозволяє моделювати просторовий рух з урахуванням тривимірного профілю поверхні пересування, так і реальних характеристик опору і зчеплення ґрунту. Однак цю модель через її складність важко використовувати на початковому етапі роботи при налагодженні моделі, націленої на відпрацювання заданих траєкторій за різної опорної поверхні з використанням адаптивного регулятора. Внаслідок необхідності використання теорії нелономних механічних систем для подальшої розробки системи управління та базового набору команд настільки складна система є незручною, тому була запропонована досить проста модель плоскопаралельного руху робота.

Рівняння динаміки плоскопаралельного руху робота:

$$\begin{aligned} m(\dot{V}_x + v\omega) &= \sum F_x, \\ m(V_x\omega - \dot{v}) &= \sum F_y, \\ J_z\dot{\omega} &= \sum M_z, \end{aligned} \tag{2.6}$$

де m – маса мобільного робота;

\dot{V}_x, V_x , – прискорення та швидкість мобільного робота у напрямку поздовжньої осі OX відповідно;

v, \dot{v} – бічна швидкість та прискорення центру мас мобільного робота відповідно;

$\omega, \dot{\omega}$ – кутова швидкість і прискорення центру мас відносно вертикальної осі OZ відповідно;

J_z – момент інерції робота щодо вертикальної осі OZ, що проходить через центр мас;

$\sum F_x, \sum F_y$ – суми проекцій на рухомі координатні осі всіх зовнішніх сил, що діють на мобільного робота з боку дороги;

$\sum M_z$ – сума моментів зовнішніх сил, що діють на мобільного робота з боку дороги, щодо вертикальної осі OZ.

Дана модель динаміки мобільного робота є досить загальною, оскільки вона описує два найважливіші режими руху робота: прямолінійний рух, у тому числі на підйомі та спуску, а також режим повороту робота.

Наприклад, вважаючи

$$\begin{aligned}v &= 0, \\ \sum F_y &= 0, \\ \sum M_z &= 0,\end{aligned}$$

отримуємо рівняння прямолінійного руху мобільного робота:

$$m\ddot{x} = P - G(\sin \gamma + f \cos \gamma), \quad (2.7)$$

де \ddot{x} – прискорення центру мас мобільного робота;

P – сила тяги;

G – сила тяжкості мобільного робота;

γ – кут підйому (спуску);

f – коефіцієнт опору коченню.

Аналогічно можна записати рівняння динаміки повороту гусеничного робота.

Рівняння динаміки плоскопаралельного руху робота доповнимо рівняннями кінематики, в яких будуть присутні доданки, що враховують явище буксування гусениць обох бортів робота:

$$\dot{x} = V \cos \alpha - v \sin \alpha - \frac{\cos \alpha}{2} (V_{\text{п}} + V_{\text{л}}), \quad (2.8)$$

$$\dot{y} = V \sin \alpha + v \cos \alpha - \frac{\sin \alpha}{2} (V_{\text{п}} + V_{\text{л}}), \quad (2.9)$$

$$\dot{\alpha} = \omega + \frac{1}{2h} (V_{\text{л}} + V_{\text{п}}), \quad (2.10)$$

де \dot{x}, \dot{y} – швидкість центру мас гусеничного мобільного робота;

α – кут між поздовжньою віссю мобільного робота та обраною віссю нерухомої системи координат;

$V_{\text{п}}, V_{\text{л}}$ – швидкість буксування лівої та правої гусениці відповідно.

Характер нерівномірного повороту гусеничного мобільного робота залежить від наступних факторів:

- зовнішніх сил опору руху;
- сил тяг на гусеницях;
- ваги гусеничного мобільного робота;
- моменту інерції мобільного роботу щодо вертикальної осі (що проходить через центр тяжіння).

Розглянемо випадок нерівномірного повороту мобільного робота горизонтальній ділянці місцевості (рис. 2.3).

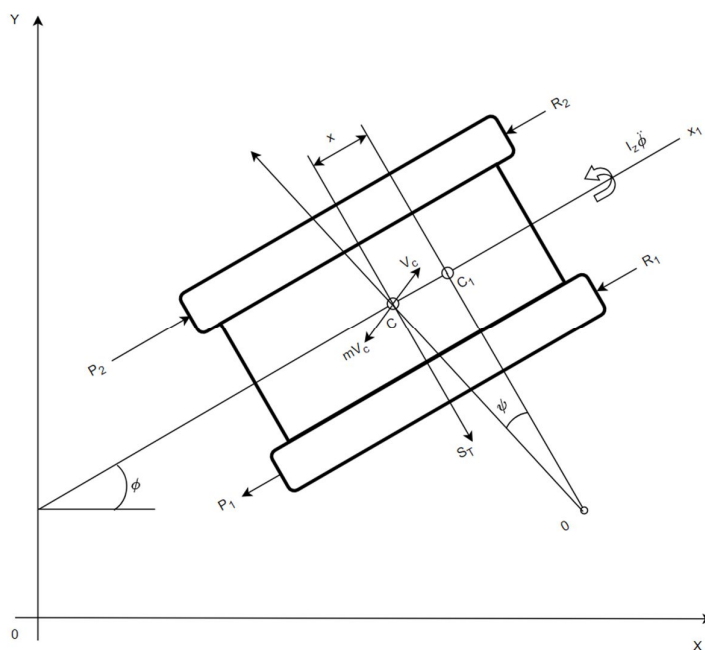


Рисунок 2.3 – Схема нерівномірного повороту гусеничного мобільного робота

Рівняння руху мобільного робота у нерухомих координатах відповідно до схеми матимуть такий вигляд:

$$\sum X = -m\ddot{x} + (P_2 - P_1 - R_2 - R_1)\cos \phi + S_T \sin \phi = 0, \quad (2.11)$$

$$\sum Y = -m\ddot{y} + (P_2 - P_1 - R_2 - R_1)\sin \phi + S_T \cos \phi = 0, \quad (2.12)$$

$$\sum M = -I_z\ddot{\phi} - M_c + (P_2 - P_1 - R_2 - R_1)\frac{B}{2} = 0, \quad (2.13)$$

де $m\ddot{x}$ – проекція сили інерції на вісь x ;

$m\ddot{y}$ – проекція сили інерції на вісь y ,

P_1 – сила гальмування гусениці, що відстає;

P_2 – сила тяги гусениці, що забігає;

R_1 – опір поступальному руху відстаючої гусениці;

R_2 – опір поступальному руху гусениці, що забігає;

S_T – результуюча поперечна сила;

M_c – момент опору повороту;

B – відстань між центрами гусениць.

Рівняння диференціальних зв'язків може бути записано наступним чином:

$$\frac{\omega_{B,K2} + \omega_{B,K1}}{2} r_{B,K} = \dot{\phi}, \quad (2.14)$$

$$\frac{\omega_{B,K2} + \omega_{B,K1}}{2} r_{B,K} = v_{сш} = \dot{x} \cos \phi + \dot{y} \sin \phi, \quad (2.15)$$

де $\omega_{B,K1}$ – кутова швидкість ведучого колеса гусениці, що відстає;

$\omega_{B,K2}$ – кутова швидкість ведучого колеса гусениці, що забігає;

$v_{сш}$ – складова швидкості v_c по поздовжній осі гусеничного мобільного робота x_1 ;

$r_{B,K}$ – радіус ведучого колеса.

Перейдемо від нерухомих координат до рухомих у системі координат, що з центром мас гусеничного мобільного робота (рис. 2.3).

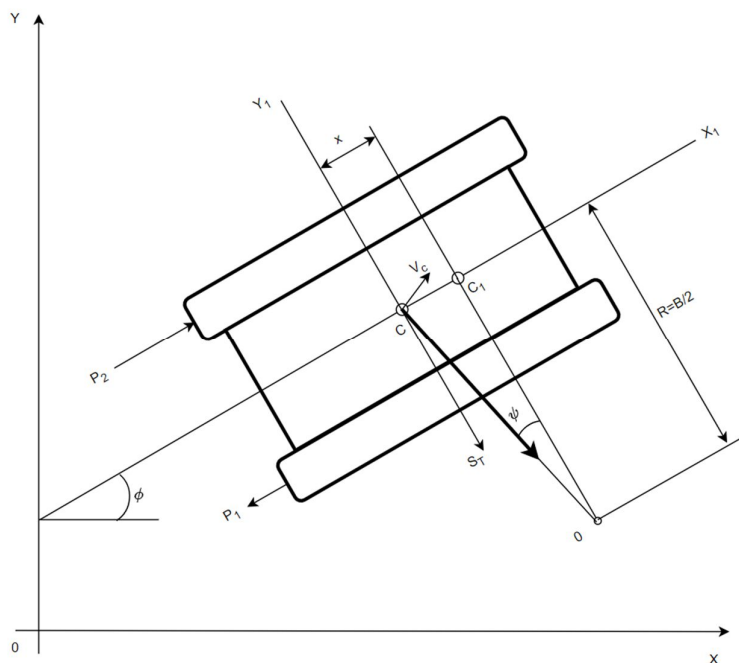


Рисунок 2.3 – Схема повороту в рухомих координатах

Рівняння руху гусеничного мобільного робота в рухомих координатах мають вигляд:

$$\sum X_1 = -m\ddot{x}_1 + P_2 - P_1 - R_2 - R_1 = 0, \quad (2.16)$$

$$\sum Y = -m\ddot{y}_1 - S_T, \quad (2.17)$$

$$\sum M = -I_z\ddot{\phi} - M_c + (P_2 - P_1)\frac{B}{2} = 0. \quad (2.18)$$

Для повного розв'язання задачі також наведено рішення рівняння кінетичної енергії гусеничного мобільного робота. Звідси можна визначити значення швидкості:

$$\dot{v}_{\text{сш}} = \frac{1}{\delta_{\text{п}} m} \left[P_{\text{д.п}} - P_1(1 - \eta_p) \frac{R-B}{R-B/2} - R_{\text{п}} \right] = \ddot{x}_1, \quad (2.19)$$

$$\ddot{\phi} = \frac{\dot{v}_{\text{сш}}}{R-B/2}, \quad (2.20)$$

де η_p – ККД ланцюга механізмів від відстаючої гусениці до тієї, що забігає;

$P_{\text{д.п}}$ – сила тяги по двигуну;

$\delta_{\text{п}}$ – коефіцієнт обліку обертових мас при повороті;

$R_{\text{п}}$ – опір руху при повороті.

На основі представлених рівнянь, що описують динаміку руху гусеничного мобільного робота, вирішують пряме завдання системи автономного керування рухом для цифрового двійника мобільної безпілотної платформи.

2.3 Висновки по другому розділу

В другому розділі кваліфікаційної роботи наведені загальні відомості про об'єкт управління. В якості об'єкта управління обрано гусеничну мобільну роботизовану платформу.

Побудовано математичну модель мобільної гусеничної моделі. Наведена форма рівнянь динаміки руху мобільного робота, записаних у рухомій системі координат. Наведені рівняння динаміки плоскопаралельного руху робота доповнимо рівняннями кінематики, в яких будуть присутні доданки, що враховують явище буксування гусениць обох бортів робота.

Розглянута схема нерівномірного повороту гусеничного мобільного робота та схема повороту в рухомих координатах.

3 АРХІТЕКТУРА АВТОМАТИЗОВАНОЇ СИСТЕМИ

3.1 Опис архітектури та принципу дії автоматизованої системи

Архітектура автоматизованої системи дистанційного керування мобільним роботом показана на рисунку 3.1.

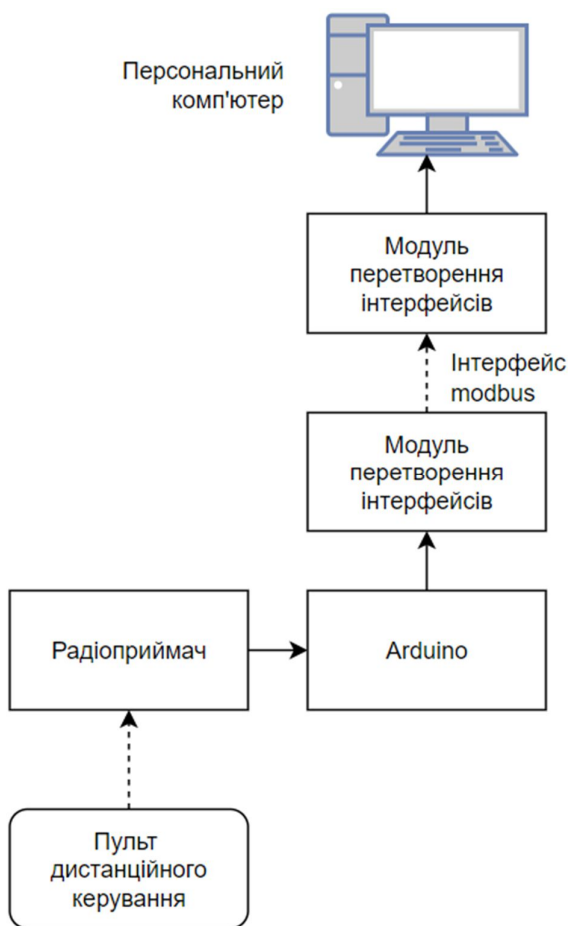


Рисунок 3.1 – Архітектура автоматизованої системи дистанційного керування мобільним роботом

До складу автоматизованої системи входять:

- пульт дистанційного керування;
- радіоприймач;
- контролер Arduino;
- модуль перетворення інтерфейсів з боку приймача;
- модуль перетворення інтерфейсів з боку персонального комп'ютера;
- персональний комп'ютер.

Як видно з рисунку 3.1 автоматизована система для дистанційного керування мобільним роботом складається з кількох основних компонентів, кожен з яких виконує свою специфічну функцію для забезпечення передачі та обробки команд.

Пульт дистанційного керування є основним засобом для передачі команд оператором мобільному роботу. Flysky FS-i4X працює на радіочастотному зв'язку 2,4 ГГц і дозволяє оператору керувати роботом шляхом зміни положення джойстиків та інших елементів управління. Пульт генерує ШІМ (широтно-імпульсну модуляцію) сигнали, які передаються на приймач.

Приймач отримує радіосигнал від пульта і перетворює його на ШІМ-сигнали, які передаються на контролер. Цей компонент є посередником між пультом дистанційного керування та іншими частинами системи.

Контролер Arduino використовується для зчитування та обробки ШІМ-сигналів з радіоприймача. Arduino має можливість підключення до кількох ШІМ-виходів, що дозволяє контролювати різні параметри роботи мобільного робота (швидкість, напрямок руху, маневрування тощо). Контролер зчитує тривалість імпульсів кожного каналу і перетворює ці дані в числові значення, які можуть бути передані через інтерфейси зв'язку.

У даній системі використовується протокол Modbus для комунікації між контролером Arduino та персональним комп'ютером. Модуль перетворення інтерфейсів відповідає за перетворення пакетів інтерфейсу UART в Modbus.

Модуль, встановлений на персональному комп'ютері, приймає пакети даних у форматі протоколу Modbus, перетворюючи їх у зрозумілі для

програми-тренажера команди. Він також забезпечує зворотній зв'язок, якщо тренажер має відправляти команди назад на контролер або мобільний робот.

ПК є основним елементом для моделювання та обробки даних. Програма-тренажер, що працює на ПК, отримує дані від Arduino через протокол Modbus, аналізує ці дані та візуалізує їх у вигляді симуляції. Це дозволяє користувачу відслідковувати рухи робота в реальному часі та тренуватись без використання фізичного обладнання.

Процес дистанційного керування мобільним роботом за допомогою даної системи можна розділити на кілька ключових етапів (рис. 3.2).

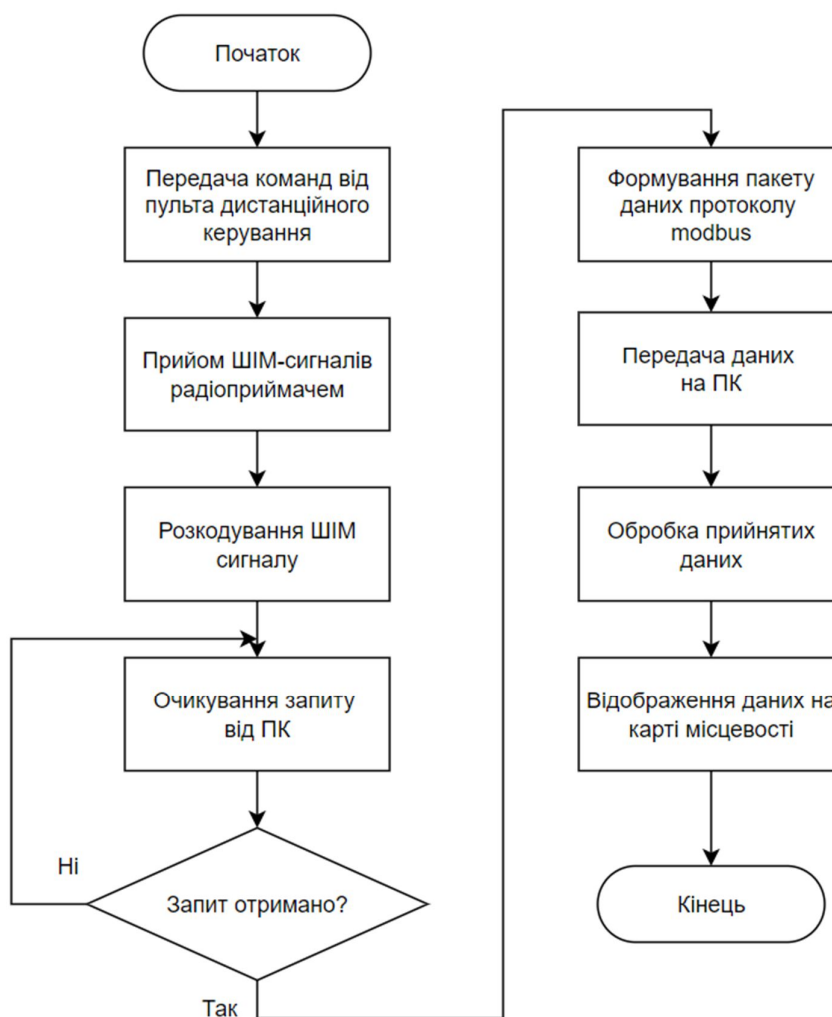


Рисунок 3.2 – Алгоритм роботи автоматизованої системи

На першому етапі відбувається передача команд від пульта дистанційного керування. Оператор використовує пульт дистанційного керування Flysky F6-i4X для генерації команд. Кожен рух джойстика або інше натискання кнопки пульта створює певну комбінацію ШІМ-сигналів, які передаються на радіоприймач.

Наступним кроком виконується прийом ШІМ-сигналів радіоприймачем. Радіоприймач отримує радіосигнали від пульта, які містять ШІМ-сигнали, що характеризують параметри керування (наприклад, напрямок, швидкість). Приймач передає ці сигнали на контролер Arduino через відповідні ШІМ-виходи.

Далі, контролер Arduino підключений до кількох ШІМ-виходів радіоприймача і постійно зчитує тривалість імпульсів на кожному каналі. Тривалість цих імпульсів відповідає певним командам керування, таким як кут повороту коліс або швидкість. Arduino обробляє ці імпульси, переводить їх у цифрові дані і готує для передачі через протокол Modbus.

Наступним кроком формуються відповідні пакети та виконується передача даних за допомогою протоколу Modbus за запитом від майстер-пристрою, в якості якого виступає персональний комп'ютер. Контролер Arduino кодує отримані ШІМ-сигнали в пакети даних, які передаються на персональний комп'ютер через протокол Modbus. Цей протокол забезпечує надійну передачу даних між пристроями та дозволяє легко масштабувати систему для додавання нових компонентів.

Програма-тренажер, встановлена на персональному комп'ютері, отримує дані від контролера Arduino та обробляє їх. Програма моделює рух робота, візуалізуючи це у вигляді симуляції на карті місцевості. Користувач може бачити, як би повадився мобільний робот при виконанні тих чи інших команд.

Перспективним розвитком даної теми є реалізація зворотного зв'язку з оператором. У випадку, якщо система підтримує зворотній зв'язок, програма-тренажер може генерувати команди, які передаються назад на Arduino для

коригування руху робота в реальному часі. Це дозволяє створити ще реалістичнішу симуляцію та тренування оператора.

3.2 Структурна схема системи автоматичного управління

Керування роботизованими безпілотними мобільними платформами пов'язано з рішенням багатьох задач. Однією з них є задача автоматичного управління. На рисунку 3.3 показана структурна схема інтелектуальної системи керування мобільним роботом.

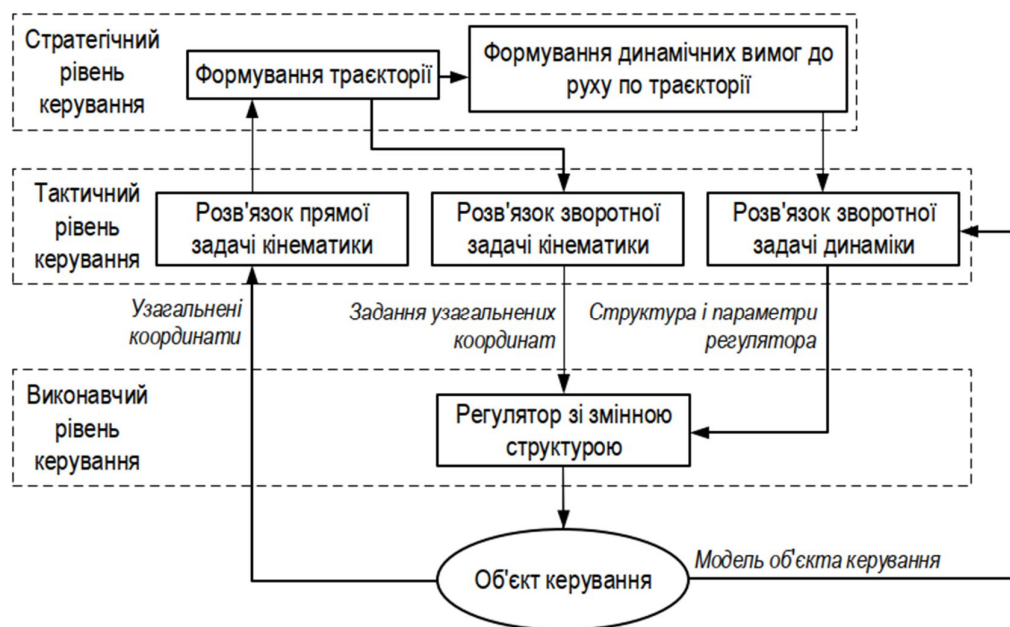


Рисунок 3.3 – Структурна схема інтелектуальної системи керування мобільним роботом

Стратегічний рівень в інтелектуальній системі керування відповідає за формування траєкторії руху робота. Основне завдання цього рівня полягає в плануванні оптимальної траєкторії з урахуванням цілей місії, навколишнього

середовища та наявних обмежень. Формування траєкторії може здійснюватися на основі картографічних даних або ж в режимі реального часу на основі даних з сенсорів (лідарів, камер, GPS).

Алгоритми планування траєкторії зазвичай спираються на методи пошуку шляху, такі як A*, Dijkstra або методи потенціальних полів. Для гусеничного робота важливо враховувати його динамічні можливості, адже він повинен не тільки досягати визначеної точки, але й робити це з мінімальним ризиком перекидання, буксування або пошкодження.

Крім того, стратегічний рівень може враховувати змінні умови на маршруті, такі як наявність перешкод, пересування інших об'єктів або зміни рельєфу. Таким чином, система постійно оновлює траєкторію, адаптуючись до поточної ситуації.

Тактичний рівень виконує завдання, пов'язані з реалізацією сформованої траєкторії. На цьому рівні вирішуються задачі кінематики та динаміки робота, що забезпечують його рух у відповідності до заданих параметрів.

Пряма задача кінематики полягає у визначенні положення і орієнтації робота у просторі на основі заданих параметрів керування (швидкостей гусениць). Гусеничний робот має специфічну кінематичну схему, де кожна гусениця може рухатися незалежно, що дозволяє йому розвертатися на місці. Алгоритм розв'язання прямої задачі кінематики враховує ці особливості та визначає нове положення робота на основі зміни швидкостей гусениць за певний період часу.

Зворотна задача кінематики полягає у визначенні необхідних швидкостей гусениць для досягнення заданої траєкторії. Цей процес є ключовим для реалізації руху за складною траєкторією, зокрема, під час маневрування у вузьких просторах або при обході перешкод. Використовуючи дані про положення і орієнтацію робота, система обчислює швидкості обох гусениць, що дозволяють рухатися в напрямку цільової точки.

Динамічна модель гусеничного робота враховує сили і моменти, що впливають на його рух. Основні фактори, які потрібно враховувати при

вирішенні задачі динаміки, включають вагу робота, інерційні характеристики, опір тертю, сили взаємодії гусениць з поверхнею тощо. Задача динаміки полягає в тому, щоб коректно врахувати ці фактори при керуванні швидкостями гусениць для досягнення стабільного і безпечного руху. Це особливо важливо при роботі на схилах або на поверхнях з різним типом покриття, де зчеплення може змінюватися.

На виконавчому рівні реалізується система автоматичного регулювання зі змінною структурою, яка забезпечує безпосереднє керування приводами гусениць і підтримку стабільного руху. Змінна структура регулятора дозволяє системі адаптуватися до змінних умов і характеристик навколишнього середовища.

Автоматичний регулятор зі змінною структурою – це інтелектуальний регулятор, який може динамічно змінювати свою конфігурацію в залежності від поточних умов. Його основне завдання полягає у підтримці оптимальних параметрів руху робота, навіть коли умови значно змінюються (наприклад, перешкоди, зміни типу поверхні або нахилу). Регулятор використовує дані з сенсорів для моніторингу стану робота і середовища, а потім змінює стратегію керування, наприклад, перемикаючи режими роботи двигунів або змінюючи параметри швидкості.

Застосування регулятора зі змінною структурою дозволяє уникнути типових проблем гусеничних роботів, таких як буксування на слизьких поверхнях або надмірне перевантаження двигунів на крутих підйомах. Він забезпечує плавність руху, стабільність і швидке реагування на зміну умов.

На рисунку 3.4 наведена функціональна схема інтелектуальної системи керування гусеничним роботом з точки зору теорії автоматичного управління.

Функціональна схема інтелектуальної системи керування гусеничним роботом з точки зору класичної теорії автоматичного управління складається з кількох основних компонентів, серед яких: об'єкт керування, регулятор, база моделей керування та блок дистанційного керування. Кожен із цих блоків

відіграє ключову роль у забезпеченні стабільного та ефективного функціонування системи керування роботом.

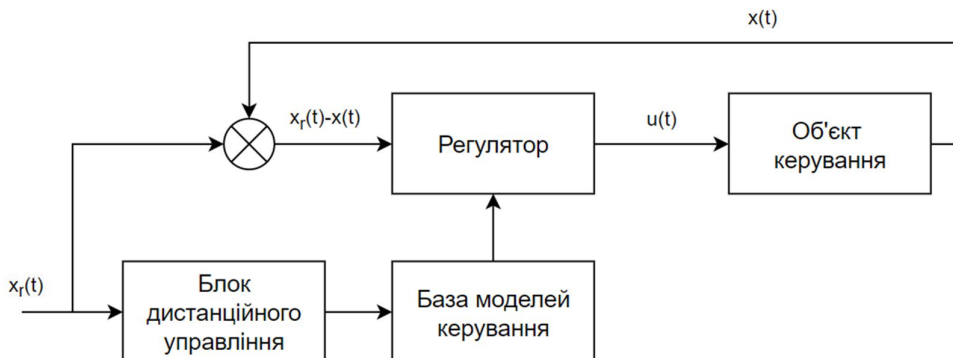


Рисунок 3.4 – Функціональна схема інтелектуальної системи керування гусеничним роботом

Об'єктом керування в даній системі є безпосередньо гусеничний робот, який виконує завдання руху відповідно до команд, отриманих від регулятора. У класичній теорії автоматичного управління об'єктом є фізична система, яка піддається впливу вхідних сигналів (команд керування) та виробляє вихідні сигнали (реальний стан системи). В нашому випадку вхідними сигналами є команди на приводи гусениць, а вихідними – положення, швидкість, орієнтація робота в просторі, які вимірюються за допомогою датчиків.

Гусеничний робот є динамічною системою з інерційними властивостями. Основні параметри, які враховуються при його керуванні, включають сили тертя, інерційні характеристики, вплив нерівностей поверхні та інші зовнішні фактори. Об'єкт керування є нелінійним, тому для забезпечення точного руху потрібен регулятор, який здатний компенсувати вплив нелінійностей і зовнішніх збурень.

Регулятор є центральним елементом функціональної схеми, оскільки він безпосередньо керує роботом, використовуючи зворотний зв'язок для стабілізації та корекції руху. У класичній теорії автоматичного управління

регулятор відповідає за підтримання стабільності системи, мінімізацію помилок і адаптацію до зовнішніх збурень.

У даній системі керування використовується регулятор зі змінною структурою, який постійно адаптується до змін у навколишньому середовищі та стані робота. Його основна функція – порівнювати задані параметри руху (запланована траєкторія, швидкість) з реальними даними від сенсорів і коригувати команди для приводів гусениць. Регулятор забезпечує стабільне досягнення цілей шляхом мінімізації відхилень між бажаним і фактичним станом робота.

У контексті класичної теорії автоматичного управління регулятор може бути пропорційно-інтегрально-диференціальним (ПІД-регулятором), який забезпечує корекцію помилок за рахунок трьох компонентів: пропорційного, інтегрального і диференціального. Змінна структура регулятора дозволяє йому адаптуватися до умов середовища, зокрема змінювати параметри коефіцієнтів для досягнення оптимального результату.

База моделей керування – це блок, який містить математичні моделі системи, що описують кінематику та динаміку гусеничного робота. Моделі є необхідними для розрахунку керуючих дій на основі зворотного зв'язку. Вони використовуються для передбачення поведінки робота на основі поточних і заданих параметрів та визначення необхідних керуючих впливів.

Відповідно до класичної теорії управління, моделі керування потрібні для розв'язання задач прямої та зворотної кінематики. Пряма кінематика дозволяє визначити положення і орієнтацію робота на основі поточних швидкостей гусениць, а зворотна кінематика – розрахувати необхідні швидкості гусениць для досягнення бажаного положення. Також у цьому блоці можуть міститися динамічні моделі, які враховують інерційні характеристики робота, вплив зовнішніх збурень та інші фактори, що впливають на рух.

База моделей постійно оновлюється залежно від змін середовища або стану робота, що дозволяє регулятору приймати більш точні рішення в реальному часі. Використання цієї бази забезпечує ефективне передбачувальне керування та мінімізацію похибок у русі робота.

Блок дистанційного керування відповідає за передачу команд від оператора або зовнішньої системи до гусеничного робота. У цій схемі дистанційне керування є важливим елементом для початкового налаштування траєкторії або для втручання в разі непередбачених ситуацій.

У класичній теорії управління цей блок може розглядатися як зовнішнє джерело сигналів, що задає референтні значення для системи. Оператор або зовнішня система задає бажану траєкторію чи інші параметри руху, які надсилаються до регулятора. Дистанційне керування може працювати в режимі ручного управління, коли оператор безпосередньо керує роботом, або в режимі автоматичного керування, де система самостійно виконує заплановані дії, коригуючи їх лише за необхідності.

Цей блок також забезпечує зворотний зв'язок для оператора, надаючи інформацію про стан робота, включаючи його положення, швидкість та інші параметри, що дозволяє вчасно реагувати на зміну умов або виникнення проблем.

3.3 Висновки по третьому розділу

Система, що описана у даному розділі, є ефективним засобом для тренування операторів мобільних роботів у режимі симуляції. Вона поєднує апаратні та програмні компоненти, такі як пульт дистанційного керування Flysky FS-i4X, контролер Arduino та програму-тренажер, що дозволяє повністю моделювати процес дистанційного керування мобільним роботом. Така система дозволяє знизити витрати на реальні тестування та підвищити якість навчання операторів у безпечному середовищі.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

4.1 Вибір компонентів для побудови макету

Відповідно до архітектури автоматизованої системи дистанційного керування мобільним роботом (рис. 3.1) виконаємо вибір компонентів для побудови макету автоматизованої системи.

4.1.1 Вибір апаратури радіоуправління

В якості апаратури радіоуправління використовується пристрій Flysky F6-i4X (рис. 4.1).

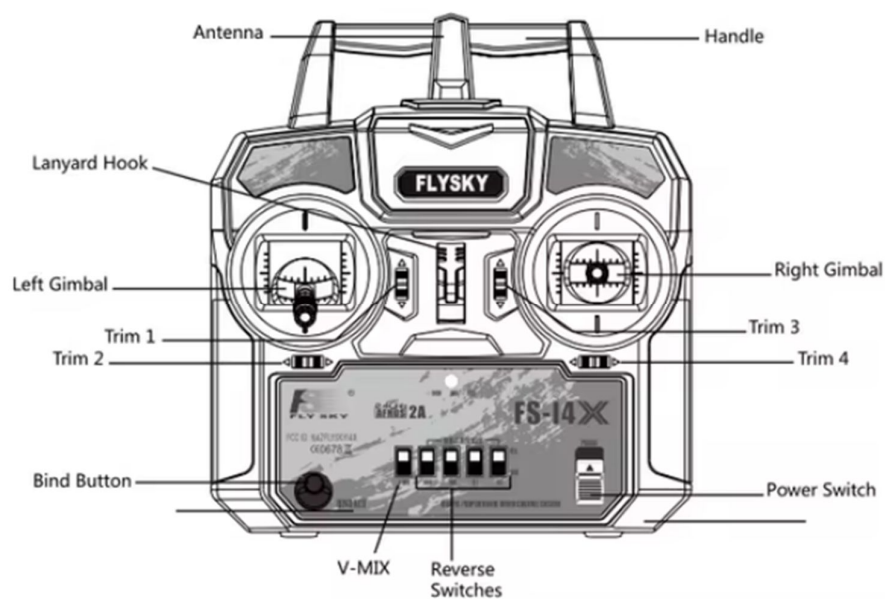


Рисунок 4.1 – Апаратура радіоуправління Flysky FS-i4X

Радіоапаратура FS-i4X є сучасною системою дистанційного керування, що широко використовується для керування моделями наземних

транспортних засобів, дронів і авіамоделей. Вона працює в діапазоні частот 2,4 ГГц і використовує технологію AFHDS 2A (Automatic Frequency Hopping Digital System), що забезпечує стабільний і надійний зв'язок між передавачем і приймачем. Ця технологія автоматично змінює частоти передачі сигналу, що дозволяє уникати перешкод і забезпечує надійність керування навіть в умовах сильної радіоінтерференції.

FS-i4X підтримує до 4 каналів керування, що дозволяє користувачам виконувати базові функції керування моделлю, такі як рух вперед-назад, повороти, а також керування додатковими функціями, наприклад, механізмами підйому або повороту. Для більш складних завдань апаратура може бути модернізована за допомогою додаткових модулів.

Приймач FS-i4X сумісний з технологією PWM (широотно-імпульсна модуляція), що забезпечує точну передачу керуючих команд до виконавчих механізмів моделі. Це робить її зручною для інтеграції з контролерами, такими як Arduino, дозволяючи легко обробляти сигнали і виконувати необхідні дії.

Радіоапаратура FS-i4X використовує кілька інтерфейсів для взаємодії з іншими пристроями, такими як моделі, контролери і зовнішні системи.

FS-i4X підтримує інтерфейс PPM для передачі багатоканальних сигналів на одному каналі. Це дозволяє передавати дані від передавача до приймача через один сигнал, що зменшує необхідність використання великої кількості кабелів.

Радіоапаратура оснащена тренувальним портом, що дозволяє підключати додаткові передавачі для навчання нових користувачів або для проведення тренувальних сесій. Це дозволяє одному оператору контролювати модель, а іншому – навчатися або коригувати управління.

4.1.2 Вибір радіоприймача

В якості приймача обрано пристрій FS-A6. Даний пристрій є частиною радіоапаратури Flysky та є популярним вибором для різноманітних моделей, таких як наземні транспортні засоби, літаки, вертольоти та дрони. Цей

приймач відзначається високою надійністю, простотою в налаштуванні та сумісністю з багатьма передавачами Flysky, зокрема з такими моделями, як FS-i4X, FS-i6X та інші, що використовують технологію 2,4 ГГц для передачі сигналу.

Зовнішній вигляд пристрою показано на рисунку 4.2.

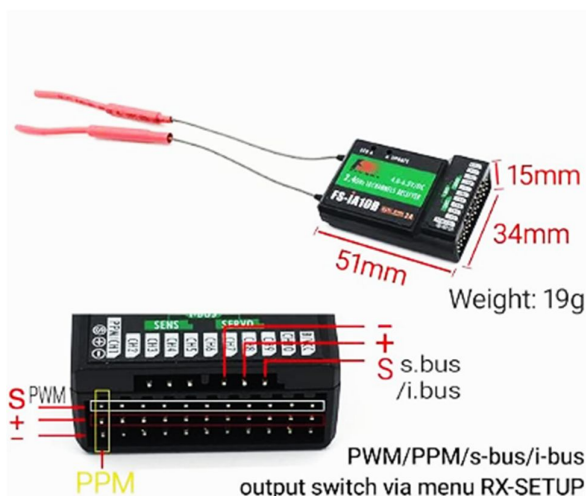


Рисунок 4.2 – Зовнішній вигляд приймача FS-A6

Приймач FS-A6 працює в діапазоні частот 2,4 ГГц, що є стандартом для сучасних систем дистанційного керування. Це дозволяє забезпечити надійний зв'язок без перешкод, а технологія AFHDS 2A (Automatic Frequency Hopping Digital System) гарантує автоматичну зміну частот у разі виникнення перешкод на одному з каналів. Це дозволяє уникнути збоїв у передачі сигналу, що критично важливо для безпеки та точності керування моделями.

FS-A6 підтримує до 6 каналів, що дозволяє здійснювати керування моделями з кількома функціями. Користувач може одночасно керувати такими параметрами, як рух, повороти, швидкість та додаткові механізми, наприклад, гіроскопи, сервоприводи чи підйомні механізми. Велика кількість каналів робить цей приймач універсальним для різних типів моделей.

Приймач використовує інтерфейс PWM (широко-імпульсну модуляцію) для передачі сигналів від приймача до виконавчих механізмів. Це дозволяє контролювати позиції серводвигунів або змінювати швидкість двигунів на основі керуючих команд. PWM є стандартом у багатьох моделях, що дозволяє забезпечити точність і зручність управління.

Також FS-A6 підтримує PPM (пульсова модуляція з багатоканальним сигналом), що дозволяє передавати дані на одному сигналі, використовуючи лише один канал для всіх керуючих параметрів. Це дозволяє зменшити кількість проводів, що підключають приймач до контролера, що є великим плюсом в системах з обмеженим простором.

Приймач FS-A6 може отримувати живлення від різних джерел, таких як батареї або інші зовнішні джерела живлення, що дозволяє використовувати його в різних типах моделей з різними вимогами до енергоспоживання.

FS-A6 має компактні розміри і легку конструкцію, що робить його зручним для встановлення в різноманітні моделі, включаючи невеликі безпілотні літаки та машини. Його можна легко інтегрувати в існуючі конструкції завдяки маленьким габаритам і невеликій вазі.

Принцип дії пристрою полягає в наступному. Приймач FS-A6 приймає сигнали від передавача через радіочастотний канал 2,4 ГГц. Сигнали кодується і передаються на приймач через технологію AFHDS 2A, що забезпечує високу стійкість зв'язку навіть у присутності перешкод. Після отримання сигналу приймач обробляє його та передає керуючі команди на відповідні виконавчі механізми через інтерфейси PWM або PPM.

Цей процес дозволяє користувачу здійснювати точне і стабільне управління моделлю, незалежно від складних умов середовища або інших перешкод.

4.1.3 Вибір мікроконтролера

В якості мікроконтролера обрано плату Arduino Nano.

Arduino Nano V3 AVR на ATmega328 – це функціональний аналог Arduino Uno, але розміщений на мініатюрній платі. Відмінність полягає у відсутності власного гнізда для зовнішнього живлення, використанням чіпа CH340 для USB-Serial перетворення та застосуванням mini-USB кабелю для взаємодії замість стандартного (постачається без кабелю). В іншому, начинка та способи взаємодії збігаються з базовою моделлю. Зовнішній вигляд плати показано на рисунку 4.3.

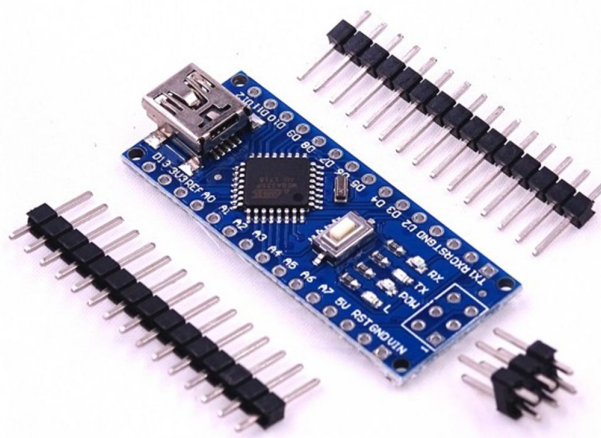


Рисунок 4.3 – Зовнішній вигляд плати Arduino Nano

Платформа має штирьові контакти, що дозволяє легко встановлювати її на breadboard. Використовуйте Arduino Nano там, де важлива компактність, а можливостей Arduino Pro Mini або не достатньо, або не хочеться займатися паянням.

Ардуїно Нано може отримувати живлення через підключення Mini-USB, або від нерегульованого 6-20 В (вихід 30), або регульованого 5 В (вихід 27), зовнішнього джерела живлення. Автоматично вибирається джерело з найвищою напругою.

4.1.4 Вибір перетворювача інтерфейсів

Зовнішній вигляд перетворювача інтерфейсів показано на рисунку 4.4.

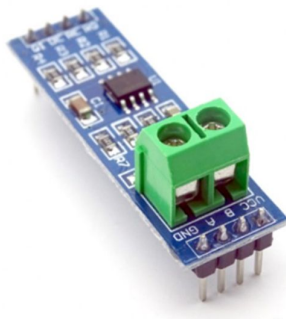


Рисунок 4.4 – Зовнішній вигляд перетворювача інтерфейсів

Перетворювач RS485 – UART(TTL) (MAX485) заснований на базі широко відомої енергоефективної мікросхеми MAX485 і дозволяє перетворювати сигнали TTL рівня на стандарт RS485 і навпаки. Крім мікросхеми та її обов'язки на платі є чотири штирьові роз'єми і двоконтактний клемник для підключення RS485, а також чотири штирьові роз'єми для TTL. У модулі реалізована можливість керування всіма виходами через мікроконтролер. Невеликі розміри модуля дозволяють створювати компактні пристрої.

4.1.5 Опис схеми підключення компонентів макету

На рисунку 4.5 показана схема поєднання компонентів макету.

Схема поєднання компонентів макету для управління гусеничним роботом за допомогою радіоапаратури FS-i4X та приймача FS-A6 побудована таким чином, щоб забезпечити передачу керуючих сигналів від пульта дистанційного керування до контролера Arduino, а також для подальшого передавання оброблених даних до зовнішньої системи через інтерфейс RS-485.

Приймач FS-A6 отримує сигнали від передавача FS-i4X на частоті 2,4 ГГц. Його канали використовуються для передачі сигналів керування, які зчитуються контролером Arduino через інтерфейс PWM (широотно-імпульсну модуляцію). До контактів D10, D9, D8, D7 контролера Arduino підключені

відповідно ШІМ-виходи 1, 2, 3, 4 приймача FS-A6. Це дозволяє Arduino зчитувати керуючі сигнали з приймача та обробляти їх для подальшого керування виконавчими механізмами робота.

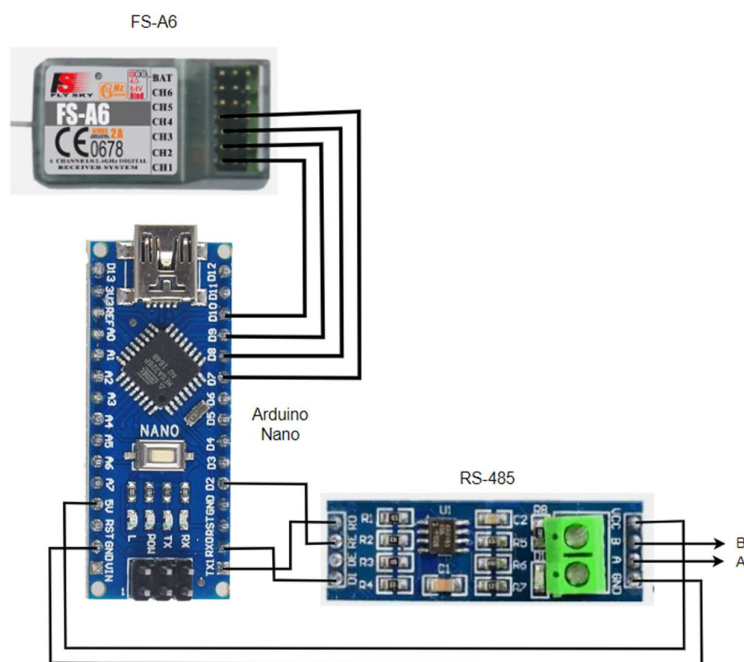


Рисунок 4.5 – Схема поєднання компонентів макету

Arduino відповідає за обробку отриманих сигналів від приймача FS-A6. Через піни D10, D9, D8, D7 контролер отримує сигнали PWM від приймача, які він може використовувати для визначення положення сервоприводів, швидкості моторів або виконання інших команд. Arduino також відповідає за передачу оброблених даних до інших систем через інтерфейс RS-485, який забезпечує зв'язок на великі відстані та стійкість до електромагнітних завад.

Перетворювач інтерфейсів RS-485 забезпечує передачу даних від Arduino до зовнішніх систем, таких як персональний комп'ютер або інші пристрої. Він підключений до пінів RX та TX Arduino, які відповідають за прийом і передачу даних через послідовний інтерфейс UART (асинхронна передача даних). Для керування режимом передачі даних використовується

вихід D2 Arduino. Цей пін контролює, чи пристрій знаходиться в режимі прийому чи передачі даних по інтерфейсу RS-485. Таким чином, Arduino може перемикатися між відправкою і прийомом сигналів, що важливо для підтримки двонаправленого зв'язку.

Комп'ютер або інша система, підключена через RS-485, може отримувати ці дані для подальшої обробки або відображення, наприклад, для моделювання в програмі-тренажері або запису даних для аналізу.

4.2 Розробка програмного забезпечення для контролера

Програмне забезпечення для контролера Arduino складається з чотирьох основних функцій:

- `setup()`;
- `loop()`;
- `checkPWM()`;
- `work_modbus()`.

Програма для контролера Arduino реалізує систему обробки сигналів від приймача радіосигналів FS-A6, що передає команди з пульта дистанційного керування FS-i4X. Контролер зчитує широтно-імпульсні модульовані (ШИМ) сигнали з шести каналів і перетворює їх на дані для передачі через інтерфейс Modbus.

На рисунку 4.6 показано фрагмент функції `setup()`.

Функція `setup()` виконується один раз при запуску програми на контролері Arduino і відповідає за ініціалізацію всіх компонентів, необхідних для подальшої роботи системи. Вона відповідає за початкове налаштування послідовного зв'язку, підключення за протоколом Modbus, налаштування пінів для прийому ШИМ сигналів і управління передачею даних, а також виконує первинну перевірку сигналів перед переходом до основної роботи системи.

```
166 void setup()
167 {
168     Serial.begin(115200);
169     slave.start();
170
171
172     pinMode(CH1, INPUT);
173     pinMode(CH2, INPUT);
174     pinMode(CH3, INPUT);
175     pinMode(CH4, INPUT);
176     pinMode(CH5, INPUT);
177     pinMode(CH6, INPUT);
178
179     pinMode(TXEN, OUTPUT);
180     pinMode(test, OUTPUT);
181
182     checkPWM();
183     digitalWrite(test, HIGH);
184 }
```

Рисунок 4.6 – Фрагмент функції setup()

Перш за все, вона запускає UART зв'язок через послідовний порт на швидкості 115200 біт/сек, що дозволяє передавати та отримувати дані між контролером і зовнішніми пристроями.

Далі функція ініціалізує роботу пристрою в режимі веденого (slave) в протоколі Modbus, що дозволяє Arduino відповідати на запити інших пристроїв через цей протокол.

Після цього контролер налаштовує пін-коди, до яких підключені виходи ШІМ приймача радіосигналів, як вхідні. Це пін-коди, що відповідають за канали ШІМ, підключені до пінів від D10 до D5 (CH1-CH6). Кожен із цих пінів зчитує сигнал ШІМ від відповідного каналу приймача FS-A6, який передає дані з пульта дистанційного керування.

Далі функція налаштовує два додаткові піни, які використовуються для керування передачею даних і тестування. Пін TXEN налаштований як вихід і відповідає за керування напрямком передачі даних через інтерфейс RS-485, а

пін test також використовується як вихідний для перевірки працездатності прогами.

Після початкового налаштування пінів викликається функція checkPWM(), яка перевіряє сигнали ШІМ від каналів приймача і зчитує їхні початкові значення. Далі, функція подає високий сигнал на пін test, для індикації того, що система готова до роботи.

Приклад функції loop() показаний на рисунку 4.7.

```
197 void loop()
198 {
199
200     pwm_CH1 = pulseIn(CH1, HIGH);
201     au16data[0] = pwm_CH1;
202
203     pwm_CH2 = pulseIn(CH2, HIGH);
204     au16data[1] = pwm_CH2;
205
206     pwm_CH3 = pulseIn(CH3, HIGH);
207     au16data[2] = pwm_CH3;
208
209     pwm_CH4 = pulseIn(CH4, HIGH);
210     au16data[3] = pwm_CH4;
211
212     pwm_CH5 = pulseIn(CH5, HIGH);
213     au16data[4] = pwm_CH5;
214
215     pwm_CH6 = pulseIn(CH6, HIGH);
216     au16data[5] = pwm_CH6;
217
218
219     slave.poll( au16data, 16 );
220     pwm_CH1 = 0;
221     pwm_CH2 = 0;
222     pwm_CH3 = 0;
223     pwm_CH4 = 0;
224     pwm_CH5 = 0;
225     pwm_CH6 = 0;
226
227 }
```

Рисунок 4.7 – Приклад функції loop()

Основна функція loop() полягає в безперервному зчитуванні ШІМ-сигналів від приймача, їх збереженні в масиві, переданні через Modbus і підготовці до наступного циклу зчитування. Функція loop() виконується

циклічно та забезпечує основну роботу програми, зокрема зчитування ШІМ-сигналів від приймача і передачу цих даних через протокол Modbus. Ось докладний опис її роботи.

Спершу, кожен з каналів ШІМ (широко-імпульсної модуляції), підключений до відповідних пінів (CH1-CH6), зчитується за допомогою функції `pulseIn()`. Ця функція вимірює тривалість імпульсу у високому стані (HIGH) для кожного з каналів. Отримане значення тривалості імпульсу зберігається у змінну `rwm_CHx` (де `x` – номер каналу).

Кожне значення ШІМ-сигналу після цього записується в масив `au16data`, який використовується для передачі даних через Modbus. Наприклад, значення тривалості імпульсу для першого каналу (`rwm_CH1`) записується в елемент `au16data[0]`, для другого каналу – в `au16data[1]`, і так далі для всіх шести каналів.

Після зчитування і збереження ШІМ-сигналів функція `slave.poll(au16data, 16)` викликає обробку даних протоколом Modbus. Вона передає масив `au16data`, що містить ШІМ-значення, іншим пристроям, які можуть запитувати ці дані через Modbus. Таким чином, програма дозволяє передавати інформацію з приймача про керуючі команди на інші пристрої або комп'ютер.

Наприкінці кожного циклу функція скидає значення змінних `rwm_CH1`, `rwm_CH2` тощо на 0, щоб підготувати їх до нового циклу зчитування.

4.3 Розробка програмного забезпечення для персонального комп'ютера

Програма для персонального комп'ютера, написана мовою C#, виконує роль тренажеру для відпрацювання навичок дистанційного керування гусеничною мобільною платформою за допомогою пульта дистанційного керування. Ця програма дозволяє оператору симулювати процес керування

мобільним роботом у віртуальному середовищі, імітуючи поведінку реального робота на основі даних, що надходять з пульта через приймач.

Основною функцією програми є симуляція руху віртуального мобільного робота по карті місцевості. Віртуальний робот керується сигналами, які передаються з пульта дистанційного керування через приймач FS-A6 до контролера Arduino. Контролер, зчитуючи ці сигнали, передає їх далі на персональний комп'ютер через протокол Modbus. Програма на ПК отримує ці дані, які кодує параметри керування рухом, та інтерпретує їх для управління віртуальним роботом на екрані.

У віртуальному середовищі програми візуалізується карта місцевості, на якій відображається маркер, що символізує мобільного робота. Оператор, використовуючи пульт дистанційного керування, може давати команди на рух робота, а маркер на екрані відповідним чином переміщується по карті. При кожному переміщенні маркеру на карті залишається слід, який відображає пройдений шлях. Таким чином, оператор може бачити траєкторію руху робота в реальному часі, що є важливим для аналізу і відпрацювання навичок управління.

Крім візуалізації, програма також дозволяє відпрацьовувати різні сценарії керування в залежності від умов місцевості, швидкості руху та траєкторії. Програма може бути налаштована для моделювання різних параметрів руху та навколишнього середовища, що допомагає оператору краще зрозуміти принципи дистанційного керування і оптимізувати навички управління.

Завдяки інтеграції з протоколом Modbus, програма здатна отримувати точні сигнали з контролера Arduino, що відповідають за конкретні параметри руху, такі як швидкість обертання гусениць, напрямок руху і положення робота на карті. Це забезпечує реалістичну симуляцію, яка максимально наближена до роботи з реальним роботом.

Зовнішній вигляд інтерфейсу програми показаний на рисунку 4.8.

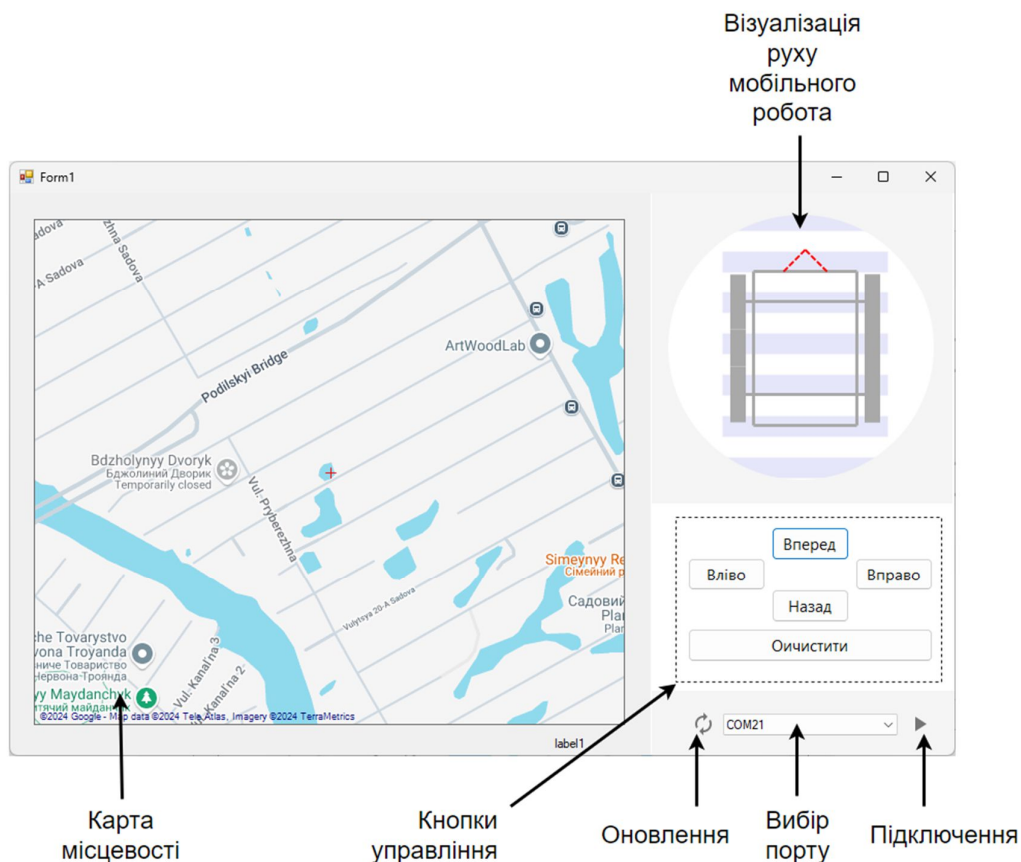


Рисунок 4.8 – Зовнішній вигляд інтерфейсу програми

В інтерфейсі програми передбачені наступні органи керування та відображення інформації:

- вікно з картою місцевості;
- кнопки локального управління;
- кнопка оновлення списку доступних послідовних портів;
- список доступних послідовних портів;
- кнопка управління підключенням до послідовного порту;
- область візуалізації руху мобільної платформи.

Вікно з картою місцевості реалізовано з використанням компоненту GMap.NET. Даний компонент – це бібліотека для роботи з інтерактивними картами, яка використовується у програмах, написаних на мові C#. Вона надає можливість інтеграції картографічних даних у додатки для відображення карт,

роботи з геолокацією та побудови маршрутів. GMap.NET підтримує різні типи карт, зокрема Google Maps, OpenStreetMap, Bing Maps, Yahoo Maps та інші, що дозволяє гнучко вибирати джерело картографічної інформації.

Основною особливістю GMap.NET є можливість інтерактивної роботи з картами, включаючи масштабування, переміщення карти, додавання маркерів та шляхів. Вона також підтримує офлайн-режим, коли дані карт завантажуються та кешуються для використання без підключення до інтернету.

Бібліотека дозволяє розробникам створювати програми для візуалізації даних на карті, що може бути корисним для проєктів, пов'язаних з моніторингом транспортних засобів, навігаційними системами, або симуляцією руху, наприклад, для моделювання пересування мобільних роботів. Вона забезпечує багатий набір функцій для взаємодії з географічними об'єктами, включаючи розрахунок відстаней, побудову маршрутів та обробку різних типів координат.

Завдяки GMap.NET, користувачі можуть легко додавати на карту маркери, які представляють конкретні місця або об'єкти, зокрема, як у випадку з віртуальним роботом у програмі-тренажері. Бібліотека дозволяє відслідковувати рух маркера та зберігати траєкторію пересування, що є важливим для багатьох застосувань, таких як моделювання маршруту або аналіз переміщень об'єктів.

Даний компонент ініціалізується на початку роботи програми завдяки наступному коду:

```
private void InitializeMap()
{
    // Ініціалізація карти
    gmap.MapProvider = GMapProviders.GoogleMap;
    gmap.Position = new PointLatLng(50.4501, 30.5234); // Київ
    gmap.MinZoom = 5;
```

```

gmap.MaxZoom = 100;
gmap.Zoom = 12;
gmap.Manager.Mode = AccessMode.ServerAndCache;
gmap.DragButton = MouseButton.Left;

// Додавання шару для маркерів та траєкторій
overlay = new GMapOverlay("overlay");
gmap.Overlays.Add(overlay);

// Ініціалізація списку точок
points = new List<PointLatLng>();

// Обробник натискання на карту (вибір початкової точки)
gmap.MouseClick += Gmap_MouseClick;
}

```

У даному коді представлена функція `InitializeMap`, яка відповідає за ініціалізацію карти в додатку, використовуючи бібліотеку GMap.NET. Спочатку вибирається постачальник картографічних даних, зокрема, Google Maps, за допомогою параметра

```
gmap.MapProvider = GMapProviders.GoogleMap.
```

Потім встановлюється початкова позиція карти за координатами широти та довготи, відповідними до місця розташування Києва – «50.4501, 30.5234».

В даній функції задаються межі масштабування карти: мінімальний рівень масштабування становить 5, максимальний – 100, а початковий рівень встановлено на 12. Для режиму доступу до даних карти використовується «ServerAndCache», що дозволяє завантажувати карти з сервера та зберігати їх у кеші для подальшого використання без інтернет з'єднання. Перетягування

карти здійснюється за допомогою лівої кнопки миші, визначеної через «gmap.DragButton = MouseButtons.Left».

Далі додається новий шар для маркерів та траєкторій, який називається "overlay", і він додається до списку шарів карти. Ініціалізується список точок «points», який буде використовуватися для зберігання координат точок на карті.

Нарешті, до об'єкта карти додається обробник події кліку мишею – «gmap.MouseClick». Цей обробник відповідає за вибір початкової точки на карті при натисканні користувачем, що дозволяє інтерактивно взаємодіяти з картою.

Код обробки натискання на кнопку має наступний вміст:

```
private void Gmap_MouseClick(object sender, MouseEventArgs e)
{
    if (isFirstClick && e.Button == MouseButtons.Right)
    {
        var point = gmap.FromLocalToLatLng(e.X, e.Y);
        points.Add(point);

        // Додавання маркера на обрану точку
        marker = new GMarkerGoogle(point,
GMarkerGoogleType.red_dot);
        overlay.Markers.Add(marker);
        isFirstClick = false; // Дозволяємо керування після вибору
початкової точки
    }
}
```

Усередині даного методу перевіряється, чи це перший клік користувача (умова «isFirstClick»), і чи було натиснуто праву кнопку миші (умова «e.Button

== MouseButtons.Right»). Якщо обидві умови виконані, за координатами кліку на карті визначається географічна точка (широта та довгота) за допомогою методу «gmap.FromLocalToLatLng», який перетворює піксельні координати кліку у географічні координати.

Отримана точка додається до списку «points», який зберігає всі вибрані точки на карті. Далі створюється червоний маркер («GMarkerGoogle») на основі обраної точки і додається до шару маркерів «overlay».

Після цього змінна «isFirstClick» встановлюється в значення «false», що означає завершення вибору початкової точки, і тепер можна продовжувати керування або виконання наступних дій на карті.

При натискання на кнопки управління рухом, або при надходження команд від пульта дистанційного керування через протокол modbus, викликається функція MoveMarker():

```
private void MoveMarker() {
    if (!isFirstClick) {
        // Обчислення зміщення по осях X і Y на основі поточного кута
        double deltaLat = stepDistance * Math.Cos(currentDirection * Math.PI /
180);
        double deltaLng = stepDistance * Math.Sin(currentDirection * Math.PI /
180);
        // Оновлення позиції маркера
        var currentPosition = marker.Position;
        var newPosition = new PointLatLng(currentPosition.Lat + deltaLat,
currentPosition.Lng + deltaLng);
        // Додавання нової точки в список і оновлення маркера
        points.Add(newPosition);
        marker.Position = newPosition;
        // Малювання траєкторії
        DrawTrajectory();    }    }
```

Функція MoveMarker відповідає за переміщення маркера на карті у випадку, якщо початкова точка вже вибрана (перевіряється за допомогою змінної isFirstClick, яка повинна бути false).

Основна задача цієї функції – обчислити нове положення маркера на основі поточного напрямку руху та кроку переміщення. Спочатку обчислюється зміщення по осях широти та довготи, використовуючи значення поточного кута руху (змінна currentDirection) та довжину кроку (змінна stepDistance). Формули для обчислення зміщення використовують тригонометричні функції: косинус для обчислення зміни по осі широти (deltaLat) та синус для осі довготи (deltaLng), при цьому кут переводиться з градусів у радіани.

Далі оновлюється позиція маркера на основі поточної позиції (currentPosition). Нова позиція розраховується шляхом додавання зміщення до широти та довготи поточного положення маркера.

Після цього нова точка додається до списку точок points, щоб зберегти траєкторію руху. Маркер також оновлює свою позицію на карті, використовуючи нові координати.

Нарешті, функція викликає метод DrawTrajectory, який відповідає за відображення траєкторії руху маркера на карті, додаючи лінію руху між точками.

Обробка команд від пульта дистанційного керування, що надходять через протокол modbus, відбувається завдяки наступній функції:

```
private void timer1_Tick(object sender, EventArgs e)
{
    byte slaveID = 1;
    ushort startAddress = 0;
    ushort numOfPoints = 6;
    ushort[] holding_register = master.ReadHoldingRegisters(slaveID,
startAddress, numOfPoints);
```

```
if (holding_register[0] < 1100 ) {
    dir = "left";
    modeRun = 2;
}
else if (holding_register[0] > 1800)
{
    dir = "right";
    modeRun = 2;
}
if (holding_register[2] < 1100)
{
    dir = "down";
    modeRun = 1;
}
else if (holding_register[2] > 1800)
{
    dir = "up";
    modeRun = 1;
}
if ((holding_register[2] < 1800) && (holding_register[2] > 1100))
{
    if ((holding_register[0] < 1800) && (holding_register[0] > 1100))
    {
        dir = "-";
        modeRun = 0;
    }
}
}
```

Функція `timer1_Tick` викликається щоразу при спрацьовуванні таймера і виконує обробку сигналів, отриманих через протокол Modbus від підключеного пристрою (ймовірно, контролера Arduino). Ця функція звертається до регістрів утримання (holding registers) для отримання даних, які використовуються для визначення напрямку та режиму руху.

Спочатку визначаються параметри запиту до Modbus-майстра: використовується «slaveID = 1» для звернення до пристрою з ідентифікатором 1, стартова адреса регістрів – 0, кількість зчитуваних точок – 6. Функція «`master.ReadHoldingRegisters`» читає значення регістрів і зберігає їх у масиві «`holding_register`».

Далі, на основі значень регістрів, проводиться обробка:

- якщо значення першого регістра (відповідного за вісь X) менше 1100, то напрям руху встановлюється як "ліворуч" (`dir = "left"`), і режим роботи змінюється на 2 (`modeRun = 2`);

- якщо значення першого регістра більше 1800, то напрямок змінюється на "праворуч" (`dir = "right"`), і режим роботи також стає 2;

- якщо значення третього регістра (ймовірно, за вісь Y) менше 1100, напрямок встановлюється як "вниз" (`dir = "down"`) і режим змінюється на 1 (`modeRun = 1`);

- якщо значення третього регістра більше 1800, напрямок встановлюється як "вгору" (`dir = "up"`) і режим також змінюється на 1.

Нарешті, якщо обидва регістри (вісь X і вісь Y) знаходяться в межах між 1100 і 1800, це означає, що рух не виконується, тому напрямок встановлюється як "-" (стоп), а режим роботи стає 0 (`modeRun = 0`).

Ця логіка дозволяє визначати рух на основі значень регістрів, що відповідають за положення осей X та Y, і контролювати напрямок та режим роботи мобільної платформи.

4.4 Приклад роботи програми

На рисунку 4.9 показано перший запуск програми та встановлення початкової точки руху гусеничного мобільного робота.

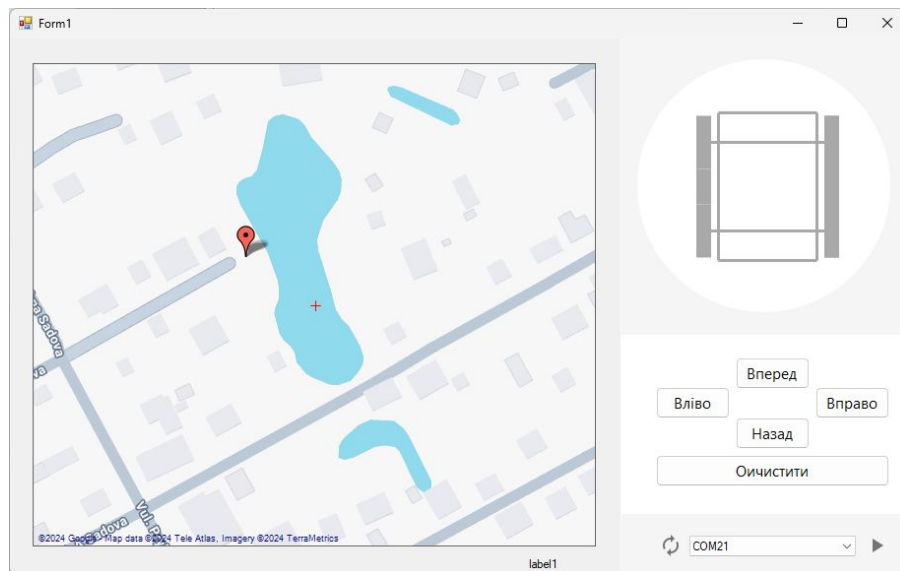


Рисунок 4.9 – Перший запуск програми та встановлення початкової точки руху гусеничного мобільного робота

Після натискання на кнопки управління віртуальний мобільний робот виконує рух заданою місцевістю. Всі його переміщення фіксуються в програмі та відображаються на екрані у вигляді траєкторії руху (рис. 4.10).

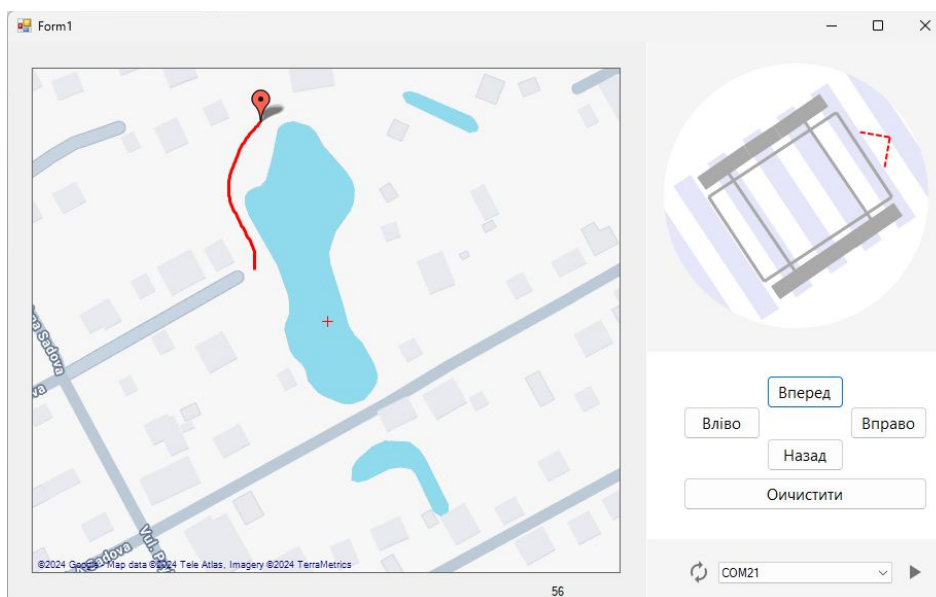


Рисунок 4.10 – Траєкторія руху мобільного робота

В області візуалізації руху мобільного робота можна бачити поточний кут обертю пристрою та напрямок руху (вперед, назад, вліво, вправо).

Так само програма дозволяє моделювати поведінку автоматизованої системи при використанні пульта дистанційного керування. Приклад сумісного застосування пульта дистанційного управління та програмного тренажеру показано на рисунку 4.11.

Програмний тренажер на ПК візуалізує процес управління в реальному часі на віртуальній моделі мобільного робота, використовуючи карту місцевості для відображення траєкторії руху. Під час керування маркер на карті, що відповідає роботу, переміщується відповідно до команд, отриманих з пульта, і формує траєкторію руху. Оператор може спостерігати за результатами своїх дій на екрані, відпрацьовуючи точність керування без необхідності використання реального робота.



Рисунок 4.11 – Приклад сумісного застосування пульта дистанційного управління та програмного тренажера

Таким чином, пульт дистанційного керування виконує роль фізичного інтерфейсу, тоді як програмний тренажер дозволяє безпечно та ефективно тренувати операторів у віддаленому керуванні мобільними роботами.

4.5 Розрахунок кількості приливу повітря з урахуванням чисельності працюючих

Розміри приміщення, в якому буде розроблятися конструкція модуля, становить 80 мм x 70 мм x 35 мм. Робоче місце складається з стола, обладнаного персональним комп'ютером (ПК). У приміщенні працює 6 осіб. Площа приміщення 56 м², обсяг – 196 м³. Згідно ДСанПіН 3.3.2.007-98 площа на одне робоче місце має становити не менше 6 м², а об'єм – 20 м³. Для даного приміщення робоча площа і об'єм на одну людину відповідає нормам, так як в нашому випадку площа на одне робоче місце становить 8 м², а обсяг – 32,6 м³.

Проведемо розрахунок кількості припливного повітря за кількістю працюючих в лабораторії людина. Відповідно СНіП 2.04.05-91 при об'ємі приміщення більше 20 м³ на одного працюючого кількість припливного повітря для провітрювання повинно бути не менше $G = 20 \text{ м}^3 / \text{год}$ на кожного працюючого [13]. Обсяг приміщення лабораторії складає 196 м³, тоді

$$V_1 = \frac{V}{n} = \frac{196}{6} = 32,66 \text{ м}^3 / \text{люд.}$$

Кількість припливного повітря з урахуванням чисельності працюючих розраховується за формулою

$$V_1 = G_1 \cdot n = 20 \cdot 6 = 120 \text{ м}^3 / \text{люд.}$$

За одну годину повітря в приміщенні навчальної лабораторії, де знаходяться люди, повинен повністю оновлюватися мінімум два рази. Якщо такого поновлення недостатньо, необхідно підрахувати, скільки ж разів потрібно проводити заміну відпрацьованого повітря на свіжий. Це називається визначенням повітрообміну по кратності

$$G = l \cdot V,$$

де l – передбачена нормативними документами кратність повітрообміну, складова для житлових приміщень 1-2, для офісних приміщень 2-3 ($l = 2$);

G – необхідна продуктивність вентиляції, м³ / год.

$$V_1 = 2 \cdot 196 = 392 \text{ м}^3 / \text{люд.}$$

Розрахувавши необхідний повітрообмін, вибираємо більшого значення кількості припливного повітря (кількість припливного повітря з урахуванням чисельності працюючих 120 м³ / год, а по кратності – 392 м³ / год), тобто в лабораторії оновлення повітря повинно відбуватися 2 рази на годину. Необхідний рівень повітрообміну досягаємо установкою вентиляційної системи з необхідними параметрами – продуктивність по повітрю не менше 400 м³ / год. Наприклад, для лабораторії вибираємо компактну припливно-втяжну установку Systemair VX 400 EV / B з продуктивністю по повітрю 400 м³ / год.

4.6 Висновки по четвертому розділу

Таким чином, програма для персонального комп'ютера, яка працює як тренажер, дозволяє операторам ефективно відпрацьовувати навички дистанційного керування мобільними платформами в умовах, максимально наближених до реальних, але без ризику для обладнання та з можливістю аналізу результатів.

Пульт дистанційного керування виконує роль фізичного інтерфейсу, тоді як програмний тренажер дозволяє безпечно та ефективно тренувати операторів у віддаленому керуванні мобільними роботами.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи розроблено програму-тренажер для моделювання поведінки гусеничного мобільного робота при дистанційному керуванні за допомогою радіоапаратури будь-якого типу. Пульти дистанційного керування виконують роль фізичного інтерфейсу, тоді як програмний тренажер дозволяє безпечно та ефективно тренувати операторів у віддаленому керуванні мобільними роботами.

В результаті виконання першого розділу кваліфікаційної роботи проведено порівняльний аналіз сфер застосування програмних тренажерів та цифрових двійників. Тренажери дозволяють імітувати роботу систем у контрольованому середовищі, що знижує ризики помилок та втрат, забезпечуючи безпечне навчання операторів.

Використання програмних тренажерів дає змогу майбутнім фахівцям вивчати нові технології в інтерактивному середовищі, що значно покращує їхню підготовку. Завдяки цим тренажерам студенти можуть виконувати індивідуальні та групові завдання, аналізувати власні помилки і вдосконалювати навички перед роботою з реальними пристроями.

Проведено аналіз методів та принципів моделювання поведінки мобільних роботизованих пристроїв з використанням цифрових двійників. наведено структуру цифрового двійника об'єкту. Надано приклад предметно-орієнтованої онтології, що включає встановлені онтології та метамоделі при створенні цифрового двійника.

В другому розділі кваліфікаційної роботи наведені загальні відомості про об'єкт управління. В якості об'єкта управління обрано гусеничну мобільну роботизовану платформу.

Побудовано математичну модель мобільної гусеничної моделі. Наведена форма рівнянь динаміки руху мобільного робота, записаних у рухомій системі координат. Наведені рівняння динаміки плоскопаралельного

руху робота доповнимо рівняннями кінематики, в яких будуть присутні доданки, що враховують явище буксування гусениць обох бортів робота.

Розглянута схема нерівномірного повороту гусеничного мобільного робота та схема повороту в рухомих координатах.

Система, що описана у даному розділі, є ефективним засобом для тренування операторів мобільних роботів у режимі симуляції. Вона поєднує апаратні та програмні компоненти, такі як пульт дистанційного керування Flysky FS-i4X, контролер Arduino та програму-тренажер, що дозволяє повністю моделювати процес дистанційного керування мобільним роботом. Така система дозволяє знизити витрати на реальні тестування та підвищити якість навчання операторів у безпечному середовищі.

Таким чином, програма для персонального комп'ютера, яка працює як тренажер, дозволяє операторам ефективно відпрацьовувати навички дистанційного керування мобільними платформами в умовах, максимально наближених до реальних, але без ризику для обладнання та з можливістю аналізу результатів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. ДСТУ 3008–2015. Звіти у сфері науки і техніки. Структура і правила оформлення. Документація. – Введ. 2015-06-22. - К.: Держстандарт України, 2015. - 31 с.
2. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка, освітньо-професійних програм: «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2023. 55 с..
3. Кафедра технології та автоматизації виробництва РЕЗ та ЕОЗ [Електронний ресурс] – Режим доступу: [www/ URL: https://tapr.nure.ua/wp-content/uploads/2020/12/buklet_17-30.pdf](http://www.tapr.nure.ua/wp-content/uploads/2020/12/buklet_17-30.pdf)
4. Основи наукових досліджень: Навч. посібник / І.Ш. Невлюдов, Ю.М. Олександров, А.О. Андрусевич, О.О. Чала. – Кривий Ріг: Криворізький коледж НАУ, 2019. – 396 с.
5. Положення про кваліфікаційну роботу здобувача вищої освіти на другому (магістерському) рівні [Електронний ресурс] : Наказ ХНУРЕ від 06 травня 2021 р. № 143. – Режим доступу: https://nure.ua/wp-content/uploads/Main_Docs_NURE/143-vid-06.05.2021-pro-vvedennja-v-dijurishennja-vchenoi-radi-universitetu.pdf.
6. Положення про протидію академічному плагіату в Харківському національному університеті радіоелектроніки [Електронний ресурс] : наказ ректора ХНУРЕ від 28.04.2017 р. № 290 // Нормативно-правова база ХНУРЕ : офіційний веб-портал. – Режим доступу: <https://nure.ua/universytet/normativno-pravova-baza#id13>. – Станом на 18.09.2024. – Назва з екрану.

7. M. Marian, F. Stîngă, M. -T. Georgescu, H. Roibu, D. Popescu and F. Manta, "A ROS-based Control Application for a Robotic Platform Using the Gazebo 3D Simulator," 2020 21th International Carpathian Control Conference (ICCC), High Tatras, Slovakia, 2020, pp. 1-5, doi: 10.1109/ICCC49264.2020.9257256.

8. Industrial Robot Simulation Software for Offline Programming - Tecnomatix. URL: <https://blogs.sw.siemens.com/tecnomatix/industrial-robot-simulation-software-for-offline-programming/> (date of access: 27.10.2024).

9. Невлюдов І. Ш. Застосування цифрових двійників технічних засобів автоматизації для розроблення програмно-технічних комплексів АСУ ТП : Навчальний посібник / І. Ш. Невлюдов, С. П. Новоселов, О. В. Сичова. – Харків: Видавництво Іванченка І. С., 2023. – 267 с.

10. J. Ren, L. Nalpantidis, N. A. Andersen and O. Ravn, "Building Digital Twin of Mobile Robotics Testbed Using Centralized Localization System," 2023 11th International Conference on Control, Mechatronics and Automation (ICCMA), Grimstad, Norway, 2023, pp. 139-145, doi: 10.1109/ICCMA59762.2023.10375050.

11. Pipeline for ontology-based modeling and automated deployment of digital twins for planning and control of manufacturing systems - Journal of Intelligent Manufacturing. URL: <https://link.springer.com/article/10.1007/s10845-021-01860-6> (date of access: 27.10.2024).

12. Akram M. W. Types of IoT Robot Locomotion. Medium. URL: <https://aws.plainenglish.io/iot-robot-locomotion-types-9fc2ee200339> (date of access: 26.12.2024).

13. Стиценко Т.Є., Пронюк Г.В., Сердюк Н.М., Хондак І.І. Безпека життєдіяльності : Навчальний посібник / Т.Є Стиценко, Г.В. Пронюк, Н.М.Сердюк, І.І. Хондак. – Харків: ХНУРЕ, 2018. – 336 с.